PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G05D 16/20 — **A1**

(11) International Publication Number: WO 93/25950

(43) International Publication Date: 23 December 1993 (23.12.93)

(21) International Application Number: PCT/US93/05542

(22) International Filing Date: 10 June 1993 (10.06.93)

(30) Priority data:
07/898,195 — 12 June 1992 (12.06.92) — US

(71) Applicant: UNIT INSTRUMENTS, INC. [US/US]; 22600 Savi Ranch Parkway, Yorba Linda, CA 92687 (US).

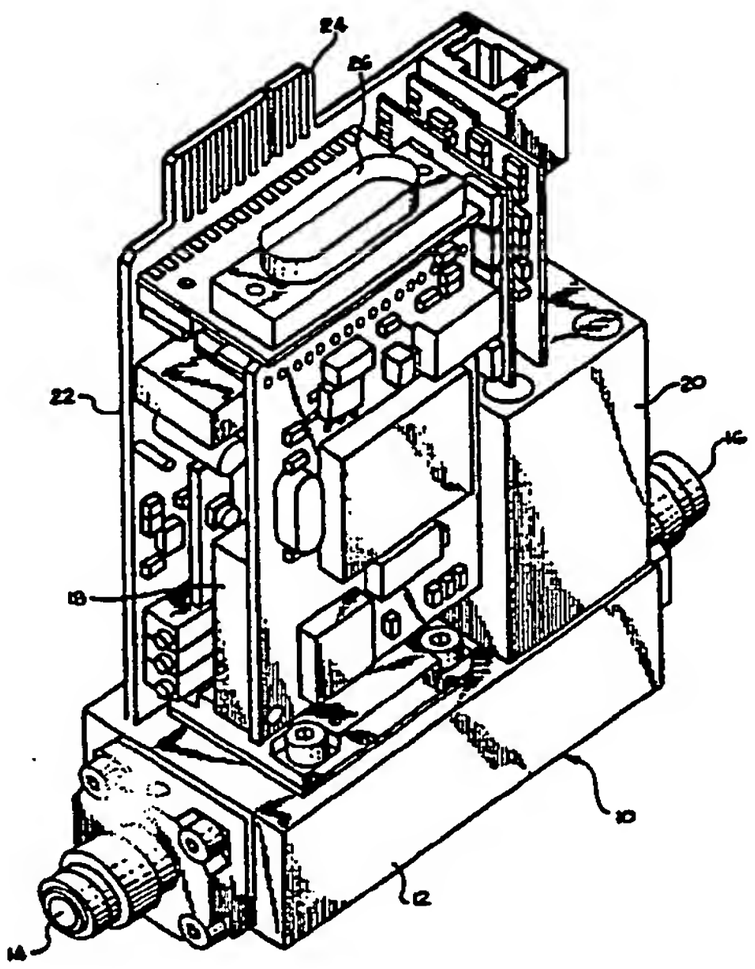
(72) Inventors: UPCHURCH, Michael, Dean ; 402 W. Francis Street, Corona, CA 91720 (US). DONALDSON, Richard, Lee ; 16677 Markham Street, Fountain Valley, CA 92708 (US). DOYLE, James, H. ; 2003 Ivy Hill Lane, Orange, CA 92667 (US). DOYLE, Michael, J. ; 10001 Platanal Drive, Villa Park, CA 92667 (US). LARAGIONE, Robert ; 2959 Amber Drive, Corona, CA 91720 (US). VU, Kim, Ngoc ; 20451 Via Marwah, Yorba Linda, CA 92686 (US).

(74) Agents: WATT, Phillip, H. et al.; Fitch, Even, Tabin & Flannery, Room 900, 135 South LaSalle Street, Chicago, IL 60603 (US).

(81) Designated States: JP, KR, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: MASS FLOW CONTROLLER



(57) Abstract

A mass flow controller (10) includes a base bypass block (12) having a flow rate sensor unit (18) connected thereto. The flow sensor (18) generates a flow signal in response to a flow of gas therethrough. A set point signal is received and combined with an internally generated set point correction signal to produce a corrected set point signal. The corrected set point signal is compared with the flow signal to produce an error signal. The error signal is used to drive a valve (20) connected to the bypass block (12) to control the flow of gas.

INSDOCID: <WO_____9325950A1_I_>

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SK	Slovak Republic
CI	Côte d'Ivoire	KZ	Kazakhstan	SN	Senegal
CM	Cameroon	LI	Liechtenstein	SU	Soviet Union
CS	Czechoslovakia	LK	Sri Lanka	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	UA	Ukraine
DK	Denmark	MG	Madagascar	US	United States of America
ES	Spain	ML	Mali	VN	Viet Nam
FI	Finland	MN	Mongolia		

INSDOCID: <WO_____9325950A1_I_>

MASS FLOW CONTROLLER

BACKGROUND OF THE INVENTION

The invention relates to a mass flow controller that may be controlled by a host process controller. More particularly, the invention relates to a mass flow controller having an internal digital linearizer.

One of the problems associated with conventional thermal mass flow controllers is that the electrical response of the thermal mass flow sensing element, which usually comprises a pair of sensing windings wound about a sensor tube through which a portion of a flow of gas travels, is nonlinear with respect to the actual mass flow rate of gas or vapor through the sensor tube.

In prior U.S. Patent No. 4,658,855 to Doyle, a mass flow controller includes a segment generator comprising a plurality of precision limiters having operational amplifiers, diodes and potentiometers. The segment generator provides piecewise linearization to the bridge output signal of the mass flow controller. Unfortunately, this type of segment generator must be adjusted by hand when the mass flow controller is manufactured and retains its hand adjustment settings even when the mass flow controller is incorporated in a gas shelf of a diffusion furnace or gas supply system of a chemical vapor deposition apparatus, or a plasma system or the like.

In addition, the segment generator only will adjust at four points along the flow curve, the 25 percent, 50 percent, 75 percent and 100 percent flow points leaving some nonlinear distortion between those points which may not be zeroed out of the mass flow controller.

Conventional thermal mass flow controllers also suffer from the disadvantage that their response may change as they age in a gas shelf, as parts are worn

3NSDOCID: <WO_____9325950A1_I_>

or as material accumulates in the sensor tube. As a result, the response of the mass flow controllers may shift while the manual settings of their linearizing segment generators do not change.

In order to relinearize such a system, it is conventionally necessary to remove the mass flow controller from the plumbed-in fittings in the gas lines of the gas shelf thereby opening the gas lines to the air which will necessitate having possibly to bake out the reassembled gas system once the mass flow controller is relinearized. This is time consuming and very expensive for a wafer fabricator. In some instances it leads to reluctance to recalibrate the thermal mass flow controllers until wafer yields are adversely affected by the changes in the thermal mass flow controllers.

What is needed then is a thermal mass flow controller that may be recalibrated in situ and which provides highly accurate response to a given flow command signal.

SUMMARY OF THE INVENTION

The present invention comprises a mass flow controller, more specifically a mass flow controller having a sensor that may respond voluntarily with respect to the rate of flow of gas. A corrected sensed mass flow signal is produced by the inventive apparatus and used as a control signal. The mass flow controller, receives a linear mass flow command or set point signal from the host process controller and produces a translated mass flow set point. Such a translated set point signal is in the form of an analog signal which conventionally may assume any value between zero and five volts with zero volts representing zero mass flow rate and five volts representing 100 percent or full flow through the mass flow controller. It is clear, for instance, that when a one volt signal, nominally representative of 20 percent flow rate, is received by the mass flow controller, the mass flow controller may

NSDOCID: <WO_____9325950A1_I_>

flow gas at a 15 percent or 25 percent mass flow rate because the mass flow controller is responding nonlinearly to the linear mass flow command signal.

The mass flow controller has stored, in a nonvolatile memory connected to a microcontroller, a multiple entry table of values which represents mass flow offsets or mass flow corrections. When the mass flow controller receives the linear mass flow command signal from the host process controller, the linear mass flow command signal is digitized by an analog-to-digital converter in the microcontroller, a processing unit of the microcontroller, in response to the flow command digitized signal, accesses a lookup table within the nonvolatile memory of the microcontroller, and the processing unit of the microcontroller generates a digitized mass flow correction signal. The digitized mass flow correction signal is fed to a digital-to-analog converter in the microcontroller. The digital-to-analog converter produces an analog flow command correction signal. The analog flow command signal is added with the linear mass flow command correction signal received from the host process controller to produce a translated or corrected mass flow command analog signal which is then compared to a corrected flow signal.

The thermal mass flow controller also includes a sensor output pin or node which generates a sensed mass flow signal which may be perturbed by the nonlinear response of the mass flow controller. The sensed mass flow signal is received by the apparatus and is digitized. The processing unit accesses a flow correction value from the EPROM. The flow correction value is supplied to the digital-to-analog converter which produces an analog sensed flow correction signal. The analog sensed flow correction signal is subtracted, in a summing amplifier, from the analog sensed mass flow signal to yield a translated or linearized mass flow

NSDOCID: <WO_____9325950A1_I_>

rate signal. The linearized mass flow rate signal is supplied to a comparator and to the host process controller. Thus, the host process controller sends a linear zero to five volt mass flow command signal and receives back a linear second mass flow rate signal in a fashion which is completely transparent to the host process controller. The signals are modified by the inventive apparatus so that a nonlinearly responding mass flow controller can operate in response to a linear signalling host process controller.

In order to generate the data which is stored in the lookup table of the EPROM a calibration system embodying the instant invention is provided. The calibration system includes means for sweeping through a set of calibration values using a sine wave signal. The sine wave signal has a period approximately ten times longer than the time constant of the mass flow controller. One of the problems which has been encountered in the past is that in order to calibrate mass flow controllers, settling time must be allowed for after each commanded flow. The settling time, for instance, for four or five points for a conventional flow controller might require a calibration time of one half hour or more.

The use of the single frequency or sine wave curve allows a large number of points, such as 256 points, to be rapidly swept through in the space of only about thirty seconds. This represents an increase of several orders of magnitude in the rate at which the mass flow controller may be calibrated. The lag primarily due to the mass flow controller time constant causes a constant phase shift to be effected because only a single frequency is being input into the mass flow controller. The phase shift may be matched by the calibrator after a first sine wave sweep. After the second sine wave sweep, the phase adjusted flow signal is compared to the setpoint on a point by point basis,

INSDOCID: <WO_____9325950A1_I_>

allowing a full 256 point calibration table to be generated in only one minute. The generated calibration table is then sent via a digital link to the mass flow controller where it is stored in the nonvolatile memory of the microcontroller. The non-volatile memory has sufficient space to store up to ten tables each of which, for instance, may be representative of the mass flow rate calibration offsets for a different gas species, e.g., nitrogen, phosphine, oxygen, phosphorus oxychloride, diborane, sulfur hexafluoride, silane and the like.

It is a principal aspect of the present invention to provide a mass flow controller which, in response to a linear command signal, generates a corrected set point signal.

It is another aspect of the present invention to provide a mass flow controller for linearizing a sensed mass flow rate signal prior to comparison to a set point signal.

Other aspects and advantages of the present invention will become obvious to one of ordinary skill in the art upon a perusal of the following specification and claims in light of the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a compensated thermal mass flow controller embodying the present invention;

FIG. 2A is a schematic diagram of a portion of the block diagram including signal summing circuits of FIG. 1;

FIG. 2B is a schematic diagram of a portion of the block diagram of FIG. 1 including a microcontroller and memory;

FIG. 2C is a schematic diagram of a portion of the block diagram of FIG. 1 including signal summing circuits;

NSDOCID: <WO_____9325950A1_I_>

FIG. 3 is a schematic diagram of a portion of the block diagram of FIG. 1 including a soft start circuit;

FIG. 4A is a schematic diagram of a portion of the block diagram of FIG. 1 including comparators for producing an error signal and a valve driven circuit;

FIG. 4B is a schematic diagram of a portion of the block diagram of FIG. 1 including an auto shutoff circuit;

FIG. 5 is a schematic diagram of a portion of the block diagram of FIG. 1 including a sensor tuning circuit;

FIG. 6A is a schematic diagram of a portion of the block diagram of FIG. 1 including a constant current source for a sensor;

FIG. 6B is a schematic diagram of a portion of the block diagram of FIG. 1;

FIG. 7 is a perspective view of the compensated thermal mass flow controller of FIG. 1;

FIG. 8 is a block diagram of a mass flow controller embodying the present invention connected in a calibration system;

FIG. 9 is a block diagram of a mass flow controller coupled to a host computer and a gas source;

FIG. 10 is a flow chart of a startup routine for the execution in the microcontroller of the mass flow controller;

FIG. 11 is a flow chart of an idle loop routine;

FIGS. 12A, 12B and 12C are flow charts of a dynamic flow correction routine;

FIG. 13A is a flow chart of a command reset routine;

FIG. 13B is a flow chart of a software set routine;

FIG. 14 is a flow chart of a command calibration routine;

INSDOCID: <WO_____9325950A1_I_>

FIG. 15 is a flow chart of a command transmit status routine;

FIG. 16 is a flow chart of command select table routine;

FIG. 17 is a flow chart of a command calibrate routine which is continued through FIG. 18;

FIG. 19 is a flow chart of a command upload routine which is continued on FIG. 20;

FIG. 21 and FIG. 22 are flow charts of a command download routine;

FIG. 23 is a flow chart of a command routine for initializing the system memory;

FIG. 24A is a flow chart of an address checking routine;

FIG. 24B is a flow chart of a system memory checking routine;

FIG. 25 is a flow chart of a calibration page verify routine;

FIG. 26 is a flow chart of a routine for transferring data from internal RAM buffers to the EEPROM in the microcontroller;

FIG. 27A is a flow chart of a command packet error routine;

FIG. 27B is a flow chart of an upload and download communication error routine;

FIG. 28 is a flow chart of a flow calibration buffer routine;

FIG. 29 is a flow chart of a check command routine;

FIG. 30A is a flow chart of an initialization routine;

FIG. 30B is a flow chart of a port initialization routine;

FIG. 30C is a flow chart of a communications initializing routine;

FIG. 30D is a flow chart of an analog-to-digital converter initialization routine;

INSDOCID: <WO_____9325950A1_I_>

FIG. 31 is a flow chart of an initialization status routine;

FIG. 32 is a flow chart of an initialization adjust routine;

FIG. 33 is a flow chart of a timer zero initialization routine;

FIG. 34 is a flow chart of a timer two disabling routine;

FIG. 35 is a flow chart of a timer one initialization routine for setting a baud rate for communications;

FIG. 36 is a flow chart of a serial IO initialization routine for disabling serial IO fort 1;

FIG. 37 is a flow chart of an initialization routine for transmission and receiving;

FIG. 38 is a flow chart of a check sum verification routine for a receive operation;

FIG. 39 is a flow chart of a check sum routine operation flow chart for verify operation;

FIG. 40 is a flow chart of a transmit buffer routine;

FIG. 41A is a flow chart of a timer routine and is continued in FIG. 41B;

FIG. 42 is a flow chart of an analog-to-digital conversion interrupt handler;

FIG. 43 is a flow chart of an analog-to-digital conversion calibration mode routine;

FIG. 44 is a flow chart of a flow mode miscellaneous address routine;

FIG. 45 is a flow chart of a UART interrupt routine;

FIG. 46 is a flow chart of a transmit byte routine;

FIG. 47 is a flow chart of a receive byte routine;

FIG. 48 is a flow chart of a break handler routine;

3NSDOCID: <WO_____9325950A1_I_>

FIG. 49 is an interrupt vector table; and

FIG. 50 is an evert vector table.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings and especially to FIG. 7, a thermal mass flow controller embodying the present invention and generally identified by numeral 10 is shown therein. The thermal mass flow controller 10 includes a bypass block 12 having connected to it an inlet port 14 for receiving a flow of process gas or vapor and an outlet port 16 for exhausting a flow of process gas or vapor. A sensor unit 18 is connected to the bypass block 12 to receive a part of the flow. A valve 20 is likewise connected to the bypass block 12 to control the flow rate of the gas. A control circuit 22 having an analog signal port 24 and a digital signal port 26 is connected to the sensor unit 18 and the valve 20. The control circuit 22 receives electrical signals from the sensor unit 18 related to the rate of flow of gas through the sensor unit 18. The control circuit 22 produces a valve driver signal for controlling the valve 20 to control the rate of flow of gas or vapor through the mass flow controller 10.

Constant current source 26 for providing current to the sensor unit 18 is shown in FIG. 6A. The constant current source has a node 28 energized by a positive 12 volt potential which in turn has a resistor 30 connected thereto. A zener diode 32 is connected to the resistor 30 and is grounded at a ground 34. A second zener diode 36 is connected to the junction of the ground 34 and the resistor 32. A potentiometer 40 is connected across zener diodes 32 and 36 and a resistor 42 is connected from potentiometer 40 and zener diode 36 to a negative 12 volt potential node 44. Regulated voltage from the resistor 30 is fed to a resistor 46 which is in turn fed to an operational amplifier 48 at its inverted terminal. The operational amplifier has a capacitor 50 connected in a feedback

NSDOCID: <WO_____9325950A1_I_>

loop from its output terminal to its non-inverting terminal. An NPN transistor 52 having a base 54, a collector 56 and an emitter 58 is connected at its base 54 to the output of the operational amplifier 48 to be driven thereby. The collector 56 is energized from a 12 volt node 60. The emitter 58 is AC coupled via a capacitor 62 to ground. A constant current line 64 is connected to the junction of the emitter 58 and to the base 62 and supplies constant current to the sensor 18, which is shown in FIG. 5. The sensor 18 includes a first temperature dependent resistance element 66 which is helically wrapped around a tube and a second temperature dependent resistance element 68 which supply a sensor output voltage at a lead 70. The resistance 68 is connected to a leads 72 which is connected to a resistor 74 coupled to the junction of the capacitor 50 and the inverting terminal of the operational amplifier 48. A capacitor 76 is connected from resistor 74 to ground. A pair of resistors 78 and 80 are connected between resistor 74 and ground. Thus, the constant current source 26 supplies a constant amount of current to the sensor unit 18 causing the same current to always flow through the resistances 66 and 68 despite changes in the flow.

A pair of taps 82 and 84 is respectively connected to leads 64 and 72. A resistor 86 is connected to lead 82, a potentiometer 88 having a sweep arm 90 is connected to the resistor 86 and resistor 92 is connected from the potentiometer 88 to the lead 84 so that the tapped voltage may be adjusted to select a nominal zero point for the sensor. That voltage is supplied to an operational amplifier 96 via a lead 98. The lead 70 is connected through a resistor 100 to a lead 102 which is also connected to the operational amplifier 106. A capacitor 104 is connected between the resistor 100 and ground. The operational amplifier 96 which comprises a preamplifier includes a capacitor 106

INSDOCID: <WO_____9325950A1_I_>

connected in a feedback loop as well as resistors 108 and 110 connected in the feedback loop and a grounded resistor 112 connected from the junction of resistors 108 and 110 to ground. The preamplifier develops an output signal at a lead 116, which is supplied to a tuning circuit 120. The tuning circuit 120 includes a resistor 122 connected to a non-inverting terminal of an operational amplifier 124. A resistor 126 and a capacitor 128 are connected in series between the resistor 122 and ground. A feedback capacitor 130 is connected in a feedback loop from the output terminal of the operational amplifier 124 to its inverting terminal. A resistor 132, a resistor 134, a resistor 136 and a resistor 138 are likewise connected in a feedback loop and a resistor 140 is connected between resistor 132 and ground. The resistor 142 is connected to potentiometer 136 and through a capacitor 144 to ground. A resistor 150 is connected to the resistor 138 and via a capacitor 162 to ground. A resistor 154 and a capacitor 156 are connected between the resistor 150 and the inverting terminal of the operational amplifier 124. An output resistor 160 is connected to the junction of the resistor 132 and the output terminal of the operational amplifier. The tuning network matches the electrical response of the sensor to other electrical portions of the mass flow controller and provides an output at a lead 162. That output of the tuned response is then supplied to an amplifier 164 which provides a full scale adjustment control comprising a potentiometer 166 connected to a grounded resistor 168. A tap resistor 170 is connected to the sweep arm of the potentiometer and to the non-inverting terminal of the amplifier. A capacitor 172 is connected between the resistor 166 and the inverting terminal of the amplifier. The amplifier 164 provides an adjusted flow output at a line 180. Line 180 is coupled to a line 182.

3NSDOCID: <WO_____9325950A1_I_>

The adjusted flow signal on line 182 is fed to a resistor 184 connected thereto which has a grounded resistor 186 connected to it. A node of the resistors 184 and 186 is connected to a summing amplifier 190 at its non-inverting terminal. The summing amplifier also receives a correction signal at its inverting terminal as will be described hereinafter. The sum of the correction signal and the adjusted flow signal provides a corrected flow signal at an output 192 which is supplied to a lead 194.

The adjusted flow signal is also fed to a low set point speedup circuit 200 having an amplifier 202 connected via a resistor 204 to the lead 182 to receive the adjusted flow signal. The amplifier includes a resistor 206 and a capacitor 208 connected in the feedback path from its output terminal to its inverting terminal at the resistor 204. A 15 volt node 210 is connected to a voltage divider comprising a resistor 212, a potentiometer 214 and a grounded resistor 216. The grounded resistor 216 is also coupled to a grounded capacitor of the amplifier 202. The amplifier 202 outputs a signal at a resistor 222 which drives a diode 224 connected thereto. A disabling circuit 230 is connected to the resistor 232 and includes a drive to the resistor 242. The transistor 242 will cause the transistor 246 to become conducting pulling the resistor 22 to ground reverse biasing the diode 224. The signal from the speedup circuit is supplied to a low pass network 250 which supplies a signal to an output amplifier 260 driving a resistor 262. The low set point speedup circuit 200 also ultimately drives a lead 270 which controls the biasing of a darlington pair of resistors used to control the valve driver circuit.

The cutoff signal from resistor 240 is also supplied to a valve driver cutoff circuit 280 comprising a first transistor 282 and a second transistor 284. Transistor 284 is connected to a lead 286 which receives

NSDOCID: <WO_____9325950A1_I_>

a valve driver signal in which when switched on will pull lead line 286 up to cutoff a darlington transistor pair driving the valve 20.

The line 239 is in turn biased by a transistor 290 as shown in FIG. 4B which is controlled from a line 292 coupled through a resistor 294 to a node 296 which is a source of negative 15 volt pull up potential. The lead 292 is also coupled via a diode network 300 to a jay pin connector 302 which when driven with an external signal through the port 24 causes the valve driver circuit to be pulled substantially down to zero whether or not .

As may best be seen in FIG. 6B, the analog input terminal 24 includes a variety of input pins included a set point input pin 330 which is coupled to a lead 332 to deliver a raw set point signal to other portions of the system. Referring now to FIG. 2C, the raw set point signal, which is received from a host computer 334, which may be a direct digital controller or the like is used to cause the mass flow controller 10 to control gas flowing from a gas source 336 such as a gas bottle through a gas line 338 connected to the inlet 14. The mass flow controller 10 responds to the signals received at its input 424, in particular the raw set point signal. However, the raw set point signal is fed via the line 332 to a summing amplifier 334 and to a resistor 336. The resistor 336 is in turn connected to a lead 338 which is connected via a lead 338 into the microprocessor 340 which in this embodiment comprises a complete microcontroller, specifically a Sygnetics 87C552 having on-board analog-to-digital conversion systems as well as pulse with modulators.

A databus 342 is connected to the micro-controller 340 and an address bus 344 is likewise connected to the microcontroller 340. The address bus and databus are connected to a latch 346 and to an EEPROM 348 for non-volatile storage of gas tables

comprising calibration data for the mass flow controller 10. The microcontroller includes a PWM0 port connected to a line 350 and a PWM1 port connected to a line 352. These lines supply pulse with modulated output signals which are meant to be respectively correction signals for the corrected set point and for the corrected flow signal. The flow signal 352 on line 352 is fed to a filter 354 which produces an effective output signal at a lead 356 comprising a correcting flow signal. That correcting flow signal is fed to an amplifier 358 which provides an amplified corrected flow signal via a resistor 360 to the flow summing amplifier 190.

Similarly, the set point signal on the lead 350 is fed to a filter 370 which provides a DC output signal on a line 372 which output signal is fed to a set point output amplifier 374. Further, the signal from the lead 372 is fed through a resistor 390 to a lead 392 and via a capacitor 394 and a resistor 396 to a lead 398.

As may best be seen in FIG. 2C, the lead 398 is coupled to a resistor 400 which is also coupled to the lead 332 feeding the summing amplifier 334 to provide a corrected set point signal at an output 402 of the summing amplifier 334. Thus, the microcontroller, as will be seen hereinafter, by accessing the lookup tables in the EEPROM, corrects both the set point and the flow signals prior to supplying them to other parts of the circuit.

Referring now to FIG. 2A it may be appreciated that a RS485 serial communications circuit, specifically a 75ALS178D referred to by numeral 450 is coupled to the microcontroller and via a pair of transceiver lines 452 and 454 to the serial port 26 for communication serially with a host computer or with a calibrator.

Referring now to FIG. 3 the corrected flow signal on line 194 is fed to a post filter 460 comprising a resistor 462 having a pair of grounded

NSDOCID: <WO_____9325950A1_I_>

capacitors 464 and 466 connected thereto. A resistor 468 is connected to the resistor 462, a resistor 470 is connected to the resistor 468 which is in turn connected to an amplifier 472 at its non-inverting terminal. The amplifier 472 has a capacitor 474 connected in a feedback loop and a resistor 476 connected to the output thereof. A resistor 478 is also connected in the feedback loop from the resistor 476 to the non-inverting terminal. A pair of capacitors 480 and 482 are connected between the resistor 470 and the resistor 476. A grounded capacitor 484 is connected to the non-inverting input terminal. The filter acts primarily as a low pass filter and provides a low pass filtered corrected flow output signal on an output lead 490. That output signal on lead 490 is also supplied to the analog connector 24 to be fed back to the host computer.

The corrected set point is fed to a buffer amplifier 500 via a resistor 502, resistor 504. A ground capacitor 506 provides low pass filtering. A resistor 510 is connected in the feedback loop of the amplifier 500. The output of the amplifier 500 drives both a direct connection 512 and a soft start system 514. The soft start system comprises a amplifier 516 connected to a combination diode network 518 and a capacitive network 520. The diode network 518 and the capacitive network 520 controls and steers current from a current source 522 and a current source 524 to provide a linear ramp voltage at a resistor 526. That ramp voltage is fed to an amplifier 528 which drives a resistor 530 to provide a ramp voltage at a lead 534. The soft start function however can be selectively enabled or disabled by a field effect transistor 536 connected via a resistor 538 to the input terminal of the amplifier 528. The transistor 536 is controlled by a signal through a resistor 540 connected to the valve off lead. When the valve off signal is supplied, the transistor 536 effectively disables the soft start

3NSDOCID: <WO_____9325950A1_I_>

system causing the corrected set point signal to be fed through line 512 to a resistor 550, also coupled to the line 534. A buffered set point line 552 is connected to the junction of line 512 and the resistor 560 to supply a buffered set point signal to an auto shut off circuit 570, which may be seen in FIG. 4B. FIG. 4B shows the buffered set point circuit which comprises a differential amplifier 572 having its inverting terminal driven by a voltage divider 574 and its non-inverting terminal receiving the buffered set point signal through a resistor 576. The output from the circuit is driven from the resistor 578 through a switching diode 580 to the line 292. Thus, if the set point is set below a certain minimum value, the differential amplifier 572 biases the line 292 to provide a positive valve off signal to other components. Such a valve off signal on the line 239, for instance, is supplied via a resistor 600 to a field effect transistor 602 which drives a resistor 604 and may be used to crowbar the soft start set point signal should soft start not be desired.

The buffered set point signal on line 552 is also supplied to a line 610 which controls a comparator shown in FIG. 4A. More specifically, the comparator is identified by numeral 612 and receives the buffered set point signal via lead 610 supplying a resistor 614 connected to its input terminal. The adjusted flow signal is fed via a line 616 connected to the line 182 through a resistor 618 to the non-inverting terminal of the amplifier 612. An output signal, which is actually the error signal, the difference between the adjusted flow signal and the corrected set point signal, is provided on line 286 to a base of a darlington pair 630 shown in FIG. 6B. The darlington pair 630 is connected to a solenoid 640 of the valve 20 and thereby controls the position of the valve as is conventional in the mass flow controller art.

VSDOCID: <WO_____9325950A1_I_>

Referring now to FIG. 10, a startup routine is shown therein which is entered when the mass flow controller 10 is started up at a step 702. The PWMP variable is set to zero and the pulse width modulator variables are set to mid-range in the step 704. Register bank zero is selected in the step 706 and RAM is initialized to zero in a step 708. In a step 710, the interrupts are disabled. In the step 712, the stack pointer is initialized. INIT ALL is called in step 714 after which a test is made in a step 716 to determine if the system memory valid flag is set. If it is, control is transferred to a step 718 where the calibration page is tested for valid. If the calibration page is valid, the DFC enable flag is tested in a step 720. If any of steps 716 through 720 are negative after testing, the idle loop is entered at step 722. If DFC enable flag is set, the receiver is enabled in step 722 and the DFC routine is entered at a step 724.

During normal execution of the system an idle loop runs as may best be seen in FIG. 10. The idle loop has an entry point at step 726, following which the process mode variable is set to zero at step 28. The address flag, STX flag, CMD flag, and CMD busy are all set to zero at step 730, and the INIT COM routine is called at step 732. The receiver is enabled at step 734. The interrupts are enabled at step 736. The routine may also be entered at a step 738 from W8_4 command if that is executed, or control is passed to a test step 740, testing to see if the command busy flag is set. If it is, the transmit busy flag is tested in step 742. If transmit is not busy, the command busy is set to zero in a step 744. Control is then transferred back to step 740. If the command is not busy, the system memory flag is checked for validity in a step 746. If it is valid, the verify calibrate page routine is called in step 748, after which the calibrate page valid is tested for in a step 750. If the system memory

3NSDOCID: <WO_____9325950A1_I_>

is not valid, control is transferred to a step 752, where communication error is tested for. If there is a communication error, the command packet error routine is accessed in step 754. If there is no communication error, receiver busy is tested for in step 756. If the receiver is not busy, command error is tested for in step 758. If there is no command error, transfer is controlled to the command vector in step 750. If there is a command error, the command packet error routine is entered in step 754.

In FIGS. 12A, 12B, and 12C, the dynamic flow correction routine is carried out. It is entered at a step 770, following which the interrupts are set in a step 772, and the 10MS flag is tested for in a step 774. If the flag is set, system memory valid step is tested in step 776. If it is not set, the FCW8 routine is entered in step 778. If the system memory is not valid in step 776, software reset is entered in a step 780. If the system memory is valid, the default page override flag is tested for in step 782. If the default page override flag is set, DPTR is set equal to default page in step 784. If it is not set, DPTR is set equal to command page in step 786. Following either step 784 or 786, the table base is stored and the DAC NO variable is refreshed. In a step 790, the 10MS flag is set to zero and FC wait is entered in step 792. FC wait is set forth in FIG. 12B where a command busy flag is tested for in a step 794. If it is busy, DFC enable is tested for in a step 796. If the command is not busy, COM ERROR is tested for in a step 798. If COM ERROR is not set, receive busy is tested for in a step 780. If there is a COM ERROR, transfer is controlled to the command packet error in a step 802. If the receive is not busy, command error is tested for in a step 804, and if there is no error, transfer is switched to the command vector routine. if the DFC enable is positive, the one millisecond flag is tested in step 808. If it

NSDOCID: <WO_____9325950A1_I_>

is negative, transfer is switched to the software reset routine in a step 710. The pin J input is tested for in a step 812. If it is negative and miscellany is busy, flag is tested for in a step 814. If it is positive, the one millisecond flag is set in the step 816 equal to zero, following which in the step 818 the DFC loop routine is entered. If the miscellaneous busy flag is not set, and auto zero enable routine is entered in a step 720 where a flag is tested for auto zero if it is positive. Auto zero time out is tested for in a step 822. If auto zero is timed out, the auto zero routine is entered in a step 824. If auto zero enable in step 820 is negative, auto zero no adjust is entered in a step 826.

The auto zero routine is substantially set forth in FIG. 12C. Following step 824, the accumulator is loaded with the flow reading. The carry is loaded with zero in a step 828. The high byte of the flow reading is tested for equal to zero in a step 830. If it is set, control is transferred to a step 832. If it is not set, control is transferred to a step 834. In step 832 if a flow reading is zero, control is transferred to a step 836 where a decrease adjustment value following a right to the pulse width monitoring ports takes place. Step 834 causes an increase adjustment to be made. If no auto zero adjust is to be made, the step 840 is executed following step 826 where the DAC NO value is written to the pulse width modulator ports.

Referring now to FIG. 13, the command reset routine is set forth therein and begins with an entry point step 850 following which the analog-to-digital converter interrupts are disabled in step 852. The TXDPTR is set equal to LBUF in step 854. The transmit count is set equal to 8 in step 856. The W8_4_TX variable is set equal to one in step 858. TX buffer is called in step 860 and control is transferred to the

3NSDOCID: <WO_____9325950A1_I_>

software reset routine in step 862. The purpose of the routine is to reply to a reset command.

Following the reply to the reset command, the software is reset as shown in FIG. 13 by the entry point at step 864. The interrupts are disabled at step 866. Register bank zero is selected at step 868 and zero is pushed twice in steps 870 and 872 following which control is transferred to the RETI routine in step 874.

A command DFC entry routine is set forth in FIG. 14 beginning with step 880 following which the analog-to-digital converter interrupts are disabled in step 882. Default page one override is set equal to one in step 884. The command page is set equal to the buffer plus an offset of two in step 886, and the verify calibration page routine is called in a step 888. After returning, the command page valid flag is tested in a step 890. If the flag is negative, control is transferred to a step 892 where the default page override and slave mode variables are set equal to zero. If the command page is valid, solo mode is disabled in a step 894 and LBUF plus two is set equal to the command page in a step 896. Following step 892, boundary errors are tested for in step 898. If there is no boundary error, control is transferred to a step 900 where the buffer at fifth position is set equal to process mode, and the LBUFFER variable at six position is set equal to system status. If there is a boundary error, the LBUFFER plus two position is loaded with binary ones in a step 902. Following step 900, the transmit pointer variable is set equal to LBUFFER to transmit count variables set equal to 8, and the W8_4_TX is set equal to one in a step 904. The TX buffer is called in a step 906. A boundary error is tested for in a step 908. If there is a boundary error, control is standard to software reset step 864. If there is none, communications are initiated by calling INIT COM in a step 910. Command flag for busy and the receive busy are set respectively to zero, and

NSDOCID: <WO_____9325950A1_I_>

one in a step 912 following which control is transferred to the DFC entry routine in a step 914.

Referring now to FIG. 15, the command transmit status routine is shown therein which initiates at a step 916. The command TXD variable is set to one. TXD valid is set to zero in step 918. TXD pointer is set to LBUF and TXD count set to 10 in step 920. The analog-to-digital converter interrupts are enabled at step 922 and a test is made to determine if miscellaneous busy flag is step in the step 924. If it is not set, load calibration buffer is called in a step 926, and W8_4_TX is called in step 928. TX buffer is called in step 930. Communication error is tested for in step 932. If there is an error, control is transferred to the software reset step 864. If there is no error, DFC enable is tested for in step 934. If DFC enable is negative, the EXEC routine is entered in step 936. If it is positive, control is transferred to DFC loop in step 938.

The purpose of the aforementioned routine is to send back current readings for the set point, the flow rate, etc., from the mass flow controller 10 to a calibrator while calibration is taking place. Such calibrator may comprise the calibrator 1000 which is connected to a source of gas 1002 and to a standard mass flow meter 1004. Gas is flowed from the gas source 1002 through the calibrator 1000, the standard mass flow meter 1004, and the mass flow controller 10 is then flowed back into the calibrator 1000. Information from the mass flow controller 10 is transferred out of the serial communication port 26 to a computer 1006 from which point it is transferred into a calibrator over a pair of busses 1008 and 1010.

In the event that a calibration page or calibration page table is to be selected, selection takes place in a command select table routine as set forth in FIG. 16. That routine is entered at a step 1020 following which the default page override is tested

3NSDOCID: <WO_____9325950A1_I_>

for in a step 1022. If the default page override is negative, default page over is set equal to one in a step 1024. The command page is set equal to DBUF plus two in the step 1026. The verified calibration page routine is called in a step 1028. If the default page override is set, variable R7 is set equal to command page in a step 1030. The command page is set equal to DBUF plus two in step 1032, and the verified calibration page is called in a step 1034. Command page valid is then tested for in a step 1036. If it is not valid, command page is set equal to R7 in a step 1038. If it is valid, control is transferred to step 1040 where W8_4_TX is set equal to one. Following step 1028 command page valid is tested for in a step 1042. If the command page is valid, control is transferred to step 1040. If it is not valid, DBUF plus two is set equal to all binary ones in a step 1044. Following step 1040 the transmit buffer routine is called in a step 1046 and communications error is tested for in a step 1048. If there is such an error, software reset is executed at a step 864. If there is no error, the DFC loop is entered at a step 1050.

Referring now to FIG. 17, the command calibrate routine is set forth therein which is executed when the calibrate mode is entered by the mass flow controller 10. The routine is entered at a step 1052 following which CAL enable is set to one in a step 1054. DFC enable is set to zero in that same step. The analog-to-digital converter interrupts are disabled in a step 1056. The TXD pointer is set in the buffer. The TXD count is set to 10 in a step 1058. The DBUF plus two contents are tested in a step 1060. If they are set, PW high and PW low are set respectively to PWM0 and PWM1 in a step 1062. If they are not set, they are set respectively to DBUF plus two, DBUF plus three, in a step 1064. In a step 1066, LBUF plus two is set to PW high, LBUF plus three is set to PW low. In a step 1068,

NSDOCID: <WO_____9325950A1_I_>

PWM0 is set to PW high, PWM1 is set to PW low. In a step 1070, TXCD valid flag is set to zero. In a step 1072, bank 3R0 is set to LBUF, bank 3R2 is set to four, bank 3R4 is set to starting analog-to-digital converter channel zero. Control is then transferred to the routine shown in FIG. 18 at a step 1074 where the analog-to-digital converter interrupts are enabled. TXCD valid flag is tested for in a step 1076, and if it is valid, the analog-to-digital converter interrupts are disabled in a step 1078. LBUF plus fourteen is then set equal to the system status in a step 1080 and W8_4_TX is set equal to one in the step 1082. The TX buffer is then called in a step 1084, and communication error is tested for in the step 1086. If there is such an error, control is transferred to software reset step 864. If there is no error RX busy is set to one and command busy is set to zero in a step 1088. Communication is then tested for again in the step 1090. If there is an error, control is transferred to the software reset step. If there is no error, receive busy is tested for in a step 1092. Control is then transferred to test for whether the command is a calibrate command. If it is, command calibrate is called in a step 1096. If it is not, the command is tested to see if it is a reset command in a step 1098. If it is, command reset is entered. If it is not, software reset is entered in a step 864.

A command upload routine is set forth in FIG. 19 which allows the microprocessor portion of the microcontroller to transmit the contents of the EEPROM to the host computer.

In that routine, the command upload step is entered at step 1102 and the analog-to-digital converter interrupts are disabled at step 1104. In the step 1106, the packet size is set to DBUF plus two, packet count to DBUF plus three, address high to DBUF plus four, and address low to DBUF plus five. Check address is called

3NSDOCID: <WO_____9325950A1_I_>

in the step 1108, and a boundary error is tested for in step 1110. If there is such an error, UL_COM_ERROR is called in a step 1112. If there is no boundary error, the packet count is tested for in step 1114. If the packet count is not greater than zero, control is transferred to step 1112. If it is greater than zero, in step 1116 DPH is set to address high, DPL is set to address low. In step 1118, R1 is set to LBUF plus two, R3 is set equal to packet size. In a step 1120, the accumulator is set equal to the address of DPTR. The address of R1 is set equal to accumulator in a step 1122. R3 is set equal to R3 minus one in a step. In other words, R3 is decremented. R3 is tested in a step 1124 to determine whether it is equal to zero. The routine continues on FIG. 19 with W8_4_TX variable is set equal to one in step 1126. The TX buff transmit buffer. The transmit buffer is called in step 1128. Communications error is tested for in step 1130. If there is such an error, UL_COM_ERROR is accessed in step 1132. If there is no error, packet count is decremented in step 1134 and is tested for in step 1136. If the packet count is zero, control is transferred to the executive in step 1138. If it is not yet reached zero, the receive buffer is initialized to zero in step 1140. Receive busy is set equal to one, and command busy to zero in step 1142. Communications error is tested for in step 1144. If there is an error, control is transferred to a step 1146 where UL_COM_ERROR is called. Receive busy is tested for in step 1146. If receive busy flag is negative in step 1146, the command is tested for to determine whether it is an upload in step 1150. If it is an upload, packet size is tested for DBUF plus two in step 1152. Packet count is tested in step 1154, address high is tested in step 1156, and address low is tested in step 1158. If all the tests are positive, control is transferred to command upload in step 1160. In the event the command is not an upload

INSDOCID: <WO_____9325950A1_I_>

in step 1150, command is tested for reset in step 1162, and if there is a command reset, command reset is accessed in step 1164.

In order to receive information from the host to the EEPROM, a command downward routine is set forth for execution on the microcontroller in FIG. 21. Command download is entered at a step 1170 following which the analog-to-digital converter is disabled at a step 1172. Packet size, packet count, address high, and address low are loaded into the data buffer vector in step 1174 and check address is called in step 1176. Boundary errors are tested for in step 1178. If there are any, DL_COM_ERROR is called in step 1180. If there are none, packet count is tested in step 1182, and if the packet count is zero, DL_COM_ERROR is called. If it is not zero, the receive pointers, transmit pointers, and transmit and receive counts are loaded in step 1184. In step 1186 W8_4_TX is set equal to one and transmit buffer is called in step 1188. The remainder of the routine is set forth in FIG. 21 following step 1188. Communication error is tested for in step 1190. If there is an error, the software reset is called in step 864. If there is no error, receive busy is set in step 1192. Communication error is tested again in a step 1194. If there is no error, receive busy is tested for in step 1196. If there is an error, DL_COM_ERROR is accessed in the step 1198. The command is tested to see if it is a download command in a step 1200. If it is, the command is used to have MEM_FILL called in a step 1202. If it is not, the command is tested for reset in step 1204. If there is a command reset, the command reset routine is called in step 1206. Following the memory fill step 1204, data compare error is measured for in a step 1208. If there is a data compare error, DL_COM_ERROR is called in a step 1210. If there is no data compare error, the data buffer is filled with the packet size, packet count, address high, address low,

INSDOCID: <WO_____9325950A1_I_>

and system status in the step 1212. W8_4_TX is set equal to one in step 1214. Transmit buffer is called in step 1216. Communication error is tested for in step 1218. Packet count is tested for in step 1220, and DLNEXT is called in step 1222.

Referring now to FIG. 23, a command initialization is for system memory is set forth therein which begins with a step 1224 following which the analog-to-digital converter is disabled in a step 1226. System memory valid is set equal to zero in a step 1228. DPTR is set equal to the system memory length. R2 is set equal to data pointer high, R3 is set equal to data pointer low in the step 1230. DPTR is then loaded with system MEM_BASE in a step 1232. ACC and R4 are set equal to zero in step 1234 and ACC accesses the data pointer in step 1236. ACC is incremented and loaded in R4 in step 1238 and R3 is decremented in step 1240. If it tested for in step 1242. R2 is decremented in step 1244 and tested for in step 1246. If it is not low, control is transferred back to step 1236. If R2 is equal to zero control is transferred to a step 1248 to determine if the right has timed out. If it has, the data pointer is set to ACC at step 1250 and the check system memory routine is called in step 1252. The LBUFFER is then filled with the status, the sum written, and the sum read in step 1254, and W8_4_TX is loaded with one in step 1256. Transmit buffer is called in step 1258. Communication error is tested for in step 1260. If there is an error, software reset is entered in step 864. System memory valid is tested for in step 1262 and the executive routine is entered in a step 1264.

In order to verify if the address is valid for data to be downloaded into a particular location, a check address routine is set forth in FIG. 24A, which is entered at a step 1266. Address high is tested to see whether it is greater than 40 in step 1268 or less than

NSDOCID: <WO_____9325950A1_I_>

20 in a step 1270. If it is greater than 40, LBUF plus four is loaded with F8 in a step 1272. If it is less than 20, LBUF is loaded with F1 in a step 1274. Boundary error is set equal to one in step 1276, and the routine returns in step 1278. The checks S memory routine is provided for in FIG. 24B which occurs upon power up at step 1280. System memory valid is set to zero in step 1282. In step 1284 DPTR is set with the system memory length. Data pointer high and low are loaded into registers 2 and 3. DPTR is then loaded with system memory base. ACC is set equal to zero and R4 is set equal to zero in step 1286. ACC is set to the address of the data pointer in step 1288. The ACC variable is incremented in step 1290, and the R3 contents are decremented at step 1292. R3 is tested in step 1294, R2 is decremented in the step 1296, and tested for in step 1298. R5 is set equal to the system memory check sum in a step 1300. Check sums are compared in step 1302, and, if equal, system memory valid is set in step 1304 following which the routine returns in step 1306. If the check sums are unequal, the system memory flag is reset in a step 1308 prior to return.

Calibration page is verified in a routine set forth in FIG. 25 beginning with a step 1310. In step 1312, calibration page valid and command page valid flags are set to zero. System memory valid is tested in step 1314. If it is not valid, return occurs in step 1316. Default page is tested in a step 1318 and calibration page valid flag is set equal to one in step 1320. Default page override is tested for in step 1322. If default page override is set in step 1324, command page valid is tested for in step 1326. Command page valid flag is loaded with one in step 1328. Boundary error is set equal to zero in the same step. If the command page is not valid, boundary error is set equal

3NSDOCID: <WO_____9325950A1_I_>

to one in the step 1330 after which routine returns in a step 1332.

In order to transfer data from the internal RAM buffer of the microcontroller to the EEPROM, the memory fill routine is provided as set forth in FIG. 26. The MEM_FILL routine is entered in a step 1334. In step 1336, ACC accesses receive data pointer is incremented by two and registers zero and seven are loaded. In step 1338 data pointer high and data pointer low are set equal to address high and address low respectively. In step 1340, ACC is set to the receive count and is decremented by three. R2 and R6 are then loaded with ACC. A loop is entered in step 1342. R0 is incremented in step 1344. Data pointer is set to the ACC in step 1346 and is incremented in step 1348. R2 is decremented in step 1350 and is tested for in 1352. In step 1354, right time out is tested. If it has timed out, R0 is set equal to R7. DPH is set to address high. DPL is set to address low in step 1358. R2 is set equal to R6 in step 1360 and ACC points to the data pointer in step 1362. ACC is then tested to determine whether it is at R0 in step 1364. If it is, the data pointer is incremented in step 1366. R0 is incremented in step 1368. R2 is decremented in step 1370 and is tested for in step 1372. In the event that the test of step 1364 is negative, data command error flag is set equal to one in step 1365 and return is entered in step 1374.

In order to test for a command packet error, a routine is set for in FIG. 27A wherein gas P stack pointer is tested to see if it is equal to stack base. If it is not, a software reset is transferred to in step 864. If it is, the executive routine step is entered in step 1380.

If either a download command error or upload command error are encountered as steps 1382 or 1384, as set forth in FIG. 27B, data buffer plus one is loaded with EE. W8_4_TX is set equal to one in step 1386.

Data buffer plus six is equal to system status in step 1388. Transmit buffer is called in step 1390, and software reset is entered in step 1392.

In order to load the calibration buffer, a load calibration buffer routine is set forth in FIG. 28 which is initiated at a step 1394. In a step 1396, the LBUFFER is filled from addresses 2 through 14 with PWM0, PWM1, set point high break, set point low break, temperature high break, temperature low break, flow out high byte, flow out low byte, valve out high byte, valve out low byte, general purpose flags, process mode, and systems status variables. In a step 1398, copy TXCD is set equal to zero and TXCD valid is set equal to one, and the routine exits at a step 1400.

The check man routine is set forth in FIG. 29 which is entered in the step 1402. Transmit busy or command busy are tested for in a step 1404. If either are busy, the routine is exited in the step 1406. If the step 1408, DBUF plus zero is tested for equality with STX. If it is not, the STX flag is set in a step 1410 following which command flag is set in a step 1412. In the event that 1408 test is positive, the STX flag is set equal to one in the step 1416. CMD is set equal to DBUF one in the step 1416. CMD variable is tested to determine whether it lies in the range 0 to 15 in a step 1418. If it does not, step 1412 is executed. If it does, command flag is set equal one in step 1420 and R2 is loaded with CMD multiplied by four in step 1422 while data pointer points to the command table. In step 1424, command busy is set equal to one and in step 1428 the routine is returned from.

A number of initialization routines are also provided. The master initialization routine is set forth in FIG. 30A and is entered at a step 1430. In a step 1432, the INIT status, INIT ports, INIT COM, INIT adjust, INIT ADC, INIT TMR0, and INIT TMR2 routines are called. In a step 1434, the start MFC INIT flag is set

INSDOCID: <WO_____9325950A1_I_>

equal to one. In a step 1436, the routine is exited. The INIT ports routine is set forth in FIG. 30B. It is entered in a step 1438. Ports P0 through P4 are loaded with all binary ones in a step 1440. Routine is exited in a step 1442. The INIT communication routine is set forth in FIG. 30C and is entered in a step 1444. In step 1446 INIT SIO1, INIT UART, and INIT TMR1 are called. The routine is exited in a step 1448. The INIT analog-to-digital converter routine is set forth in FIG. 30D and is entered in a step 1450. In a step 1452 bank 30 is set equal to miscellaneous analog, bank 3 register 1 is set equal to zero, bank 3 register 2 is set equal to four, bank 3 register 3 is set equal to 8, as are bank 3 register 4 and 5, bank 3 register 6 is set equal to table base high, bank 3 register 7 is set equal to table base low. In step 1454, ADCON is set equal to zero. The routine is exited in step 1456.

The INIT status routine previously referred to is set forth in FIG. 31 and is entered at a step 1460. The start MFC flag is tested for in step 1462. If it is not set, ACC is loaded with system status in step 1464, and ACC is tested for zero in step 1466. If ACC is not zero, the process mode timer status, EIA status, COM flags, command flags, and general purpose flags are all set equal to zero in step 1468. Check system memory is called in step 1470. System memory valid flag is tested for in 1472, and, if not valid, process mode equals zero is set in 1474 following which the routine is exited in step 1476. If the system memory is valid, verified calibration page is called in step 1478. Calibration page valid is tested for in step 1480. If it is not valid, solo mode flag is set to zero in step 1482. If it is valid, solo mode is set to one and DFC enable is set to one in step 1484. Default page override is tested in step 1486. Command page valid is tested in step 1488 following which default page override is tested again at step 1490. If the command page or

3NSDOCID: <WO_____9325950A1_I_>

default page are not valid, the slave mode is set to zero and DFC enable set to zero in step 1492. If they are both valid, slave mode is set to one, as is DFC enable in step 1494. Auto zero range error flag is tested for in step 1496. If there is no error, auto zero enable is set to one in step 1498, and the routine is exited in step 1500.

The INIT adjust routine is set forth in FIG. 32 and initiates with a step 1502. System memory flag is tested for validity in step 1504, and, if it is valid, DPTR, ACC, and PWH are all loaded in step 1506. DPTR is incremented in step 1508 and the pulse width modulator variables are loaded in step 1510. If the system memory is not valid, the pulse width modulator variables are set equal to mid-range in a step 1512. The route is returned from in a step 1514.

In order to establish a real time clock function within the microcontroller, the timer must first be initialized. The initialization routine is carried out as set forth in FIG. 33 where the routine is entered in the step 1516 following which the R0 register and bank 2 set equal to 4, R1 to 10, and R2 to 100 in step 1518. TMOD variable is loaded in step 1520. Timer zero high is set at step 1522, as is timer zero low. Timer is started in step 1524 and an interrupt is enabled at step 1526 after which the routine returns in the step 1528. The second software timer is disabled in the routine set forth in FIG. 34. That routine is entered in a step 1530 following which the variables of interest are disabled at step 1532, and the routine is exited in step 1534.

In FIG. 35, a bard rate generator timer is initialized and started beginning with step 1536. In step 1538, the TMOD variable is loaded. In step 1540, SMOD is divided by 16. In step 1542, timer high and low variables are set, and in step 1544 the time is started. The routine is exited in step 1546.

3NSDOCID: <WO_____9325950A1_I_>

The SIO function is disabled in the INIT SIO routine in FIG. 36. Beginning with step 1548, the routine is entered. The variables are disabled in step 1550. The routine is exited in step 1552.

In FIG. 37, the universal asynchronous receiver transmitter function is initialized. That is the function that controls the serial transmission through the port 26. The routine is entered in a step 1554 following which in step 1556 the receive and transmit pointers and counters are loaded. Registers 0 through 3 are then loaded with those values in step 1558. Start MFC initialization flag is tested for in step 1560. If it is set, UART mode is set to one in step 1562, the receiver is enabled, and the UART interrupts are enabled, following which in the step 1564 the W8_4_BRK and W8_4_ADR flags are set. The routine is exited in step 1566. In the event that step 1560 test is negative, system memory is tested in step 1562. If it is not valid, the RS485 address variable is loaded with all ones in step 1564, and control is transferred to step 1562. Otherwise, the data pointer is loaded with the switch address in step 1566, and the ACC value is tested for in step 1568, following which the RS485 address is loaded in step 1570.

In order to insure the accuracy of the received and transmitted data, several routines are provided. In FIG. 38, a verify receive check sum routine begins with a step 1568 following which R0 is loaded with the receive pointer, R2 is loaded with the receive counter less one, ACC is set equal to zero in step 1570, and check sum error is set equal to zero. The ACC and R0 variables are incremented, R2 variable is decremented in step 1572, and R2 is tested in step 1574. The ACC is tested for equality with check sum in step 1576. If it is not, error is flagged in step 1578 and routine is entered in step 1580. If there is no check

NSDOCID: <WO_____9325950A1_I_>

sum error, step 1582 is entered, following which the routine returns.

The transmit check sum may be calculated as set forth in FIG. 39. The routine is entered in step 1582. R1 is set equal to transmit pointer, R3 to transmit counter minus one, and ACC to zero in step 1584. ACC is added to the contents of R1, R1 is incremented, and R3 is decremented in step 1586. R3 is tested in step 1588. If it is zero, check sum is equal to ACC in step 1590, and the routine returns in step 1592.

In order to send out bytes of information to the host, a transmit buffer routine is set forth indicated in FIG. 40. The routine is entered in step 1594. The UART interrupts are enabled in step 1596. Transmit check sum routine is called in step 1598, and the bank 1 registers 1 and 3 are loaded with the transmit pointer and count while the transmit busy flag is set equal to one in step 1600. Break detect is tested for in step 1602. If it is positive, the transmitter is enabled in step 1604, and zeroes are transmitted in step 1606. The TXPC address is set equal to one in step 1608 and W8_4_TX is tested in step 1610. If it is positive, transmit busy is tested in 1612. If it is negative, the routine returns in step 1614.

The real time timer functions are initiated and provided in the routine set forth in FIGS. 41A and 41B. The routine is entered in the step 1616 following which register bank 2 is selected, ADC conversion is initiated, and timer tick is set equal to one in step 1618. Right R0 is decremented in step 1620 and is tested for in 1622. It is set equal to four in step 1624 and one MS is set equal to one in the same step. DFC enable is tested for in step 1626. Calibration enable is tested in step 1628. If it is negative, select register bank is accessed in step 1630, and routine is exited in step 1632. In the event that the

3NSDOCID: <WO_____9325950A1_I_>

DFC is enabled, the pin J flag is done in step 1634. If pin J is enabled, set point is selected in step 1636. If not, auto zero timer is enabled in step 1638 and flow is selected in step 1640, following which auto zero timer is enabled in step 1642, and R4 is set equal to auto zero time. R1 is decremented in step 1644 and is tested for in step 1646. R1 is loaded with 10 and R2 is decremented in step 1648. R2 is tested in step 1650. R2 is set equal to 100 in step 1652. The auto zero timer enable is tested for in step 1654. R4 is decremented in step 1656 and is tested for in step 1658. Auto zero timer enable is set to zero, auto zero time out flag is set to one in step 1660, after which control is transferred back to step 1630. In order to handle the analog-to-digital converter interrupts, the routine is provided as set forth in FIG. 42 which routine is entered in a step 1662. The register bank 3 is selected in step 1664. DFC enable is tested in step 1666. If the flag is not set, calibration enable is tested in step 1668 and calibration mode ADC routine is entered in step 1670. If the DFC is enabled, miscellaneous convert flag is tested in step 1672 and FM miscellaneous is called in step 1674. If miscellaneous convert flag is not set, flow command is tested in step 1676, and, if positive, ACC is loaded and R1 is set equal to ACC times two. P2 is loaded with ACC in step 1686. PWM0 and PWM1 are then loaded in step 1688 through 1702. R2 is decremented and tested for zero in step 1704 and 1706. In the event that the CAL enable flag is negative in step 1668, the analog-to-digital converter is reset in 1710. Register bank 0 is selected in 1712 and the routine is exited in step 1716.

In order to enter the calibration mode for the analog-to-digital converter, the calibration mode ADC routine is provided in FIG. 43, which is entered in a step 1720. ACC is set equal to ADCON, R3 to ADC high in step 1722. ACC data is formatted in step 1724 and R0 is

NSDOCID: <WO_____9325950A1_I_>

incremented in step 1726. ACC is set equal to six in step 1728, and R0 and R4 are both incremented. R2 is decremented in step 1730 and is tested for in step 1732. In step 1734, transmit command valid is set, R0 is loaded with LBUFFER plus four, R2 is set equal to four, R4 is set equal to eight, and control is transferred back to step 1710.

In order to step through the flow mode conversions for each of the analog-to-digital converter channels on the microcontroller, an incrementing routine FM_MISC is provided as set forth in FIG. 44. In a step 1738, the analog-to-digital channel is loaded into R0, R0 is incremented by one in step 1740. The ADC load channel is then loaded. R0 is incremented again in step 1744. R3 is decremented in step 1746 and is tested in step 1748. In step 1750, miscellaneous busy flag is set to zero, R0 is set to miscellaneous analog, R3 is set to eight, R4 is set to seven. In step 1752, R5 is incremented, and miscellaneous convert is set to zero. The UART interrupt routine is provided as set forth in FIG. 45 which begins with a step 1754, following which register bank 1 is selected in step 1756. BRK_DETECT flag is set equal to zero in step 1758, and the transmitter interrupt is tested for in step 1760. If there is an interrupt, transmit byte is called in step 762. If not, receiver interrupt is tested for in step 1764. If there is a receiver interrupt, receiver byte is called in step 1766. The UART is reset in step 1768. W8_4_BRK flag is set equal to one in step 1770. The UART interrupt is enabled in step 1772. Register bank 0 is selected in step 1774, and the routine is exited in step 1776.

The transmit byte routine is set forth in FIG. 46 and is initiated in step 1778. TI is set equal to zero in step 1780. The W8_4_BRK flag is tested in step 1782. If it is positive, the transmit break byte flag is tested in step 1784, and the W8_4_BRK flag is

3NSDOCID: <WO_____9325950A1_I_>

set equal to zero in step 1786 following which the transmit break byte flag is set equal to zero in step 1788. Break detect is set equal to one in step 1790 and UI exit is called in step 1792. In the event step 1782 is negative, a step 1794 is entered where more to send is tested for. If the flag is positive, R1 is incremented, R3 is decremented in step 1794, and UI exit is called in step 1796. If it is negative, W8_4_BRK flag is set in step 1798. Receive busy is set and transmit busy is reset following which the break handler is called in step 1800.

The receive byte routine where the UART interrupt is set forth in FIG. 47 beginning at step 1802, RI is set equal to zero in step 1804, W8_4_BRK is tested in step 1806, and, if positive, the break handler is called in step 1810. If negative, ACC is set equal to S0 buffer and W8_4_ADR is tested in step 1814 which, if positive, ACC is tested for as the RS485 address in step 1816. If step 1814 is negative, step 1820 is entered and W8_4_STX is tested for which, if negative, transfers control to step 1826 where R0 is loaded with ACC and address and then incremented. R2 is decremented in step 1828 and is tested for in step 1832 following which UI is exited in step 1834. In the event that step 1816 result is positive, step 1818 is entered and a W8_4_ADR flag is set equal to zero. Address flag is one, W8_4_STX is set equal to one, following which the variables are loaded in step 1824, if the result of step 1822 is positive. In the event the result of step 1816 is negative the W8_4_BRK and W8_4_ADR flags are set equal to one in step 1836 and the routine is exited in step 1838.

The break handler is set forth in FIG. 48. The routine is entered in step 1840. The RS485 or serial output at port 26 is disabled in step 1842. Transmit break byte flag is set in step 1850. The SO

NSDOCID: <WO_____9325950A1_I_>

buffer is loaded with zeroes in step 1852 and the routine is exited in step 1854.

While there have been illustrated and described particular embodiments of the present invention, it will be appreciated that numerous changes and modifications will occur to those skilled in the art which fall within the true spirit and scope of the present invention.

3NSDOCID: <WO_____9325950A1_I_>

WHAT IS CLAIMED IS:

1. A mass flow controller, comprising

an inlet for receiving a flow of fluid;

a bypass connected to receive the flow of fluid from said inlet;

means connected to the bypass for sensing a rate of flow of the fluid and producing a fluid flow signal in response thereto;

means for receiving a set point signal;

means for storing a correction table of fluid flow values;

means for generating a fluid flow set point correction signal for a selected fluid flow rate;

means for comparing said fluid flow set point corrected value to the fluid flow signal and producing in response thereto an error signal;

means for generating a valve command signal in response to the error signal;

a valve connected to the bypass and receiving the flow of fluid and the valve control signal and responding to the valve control signal control the rate of flow of the fluid; and

an outlet connected to receive the flow of fluid from said valve.

2. A mass flow controller as defined in claim 1 wherein said means for sensing fluid flow comprises a pair of thermally responsive elements, one of which is positioned upstream of the other.

3. A mass flow controller as defined in claim 1 wherein said means for storing a correction table of fluid flow values further comprises means for storing multiple correction tables of fluid flow values.

4. A mass flow controller as defined in claim 1 further comprising means for communicating with a calibration system for receiving data signals for storage in said correction table.

NSDOCID: <WO_____9325950A1_I_>

5. A mass flow controller as defined in claim 1 further comprising means for tuning said sensing means.

6. A mass flow controller as defined in claim 1 wherein said correction table comprises multiple entries of fluid flow correction values.

7. A mass flow controller as defined in claim 1 wherein said means for generating a fluid flow correction signal further comprises means for generating a digital fluid flow correction signal and means for converting said digital fluid flow correction signal to a analog fluid flow correction signal.

8. A mass flow controller as defined in claim 1 further comprising means for generating a constant current for energizing said sensing means.

9. A mass flow controller as defined in claim 1 wherein said sensing means comprises a pair of helically wound thermally responsive elements.

3NSDOCID: <WO_____9325950A1_I_>

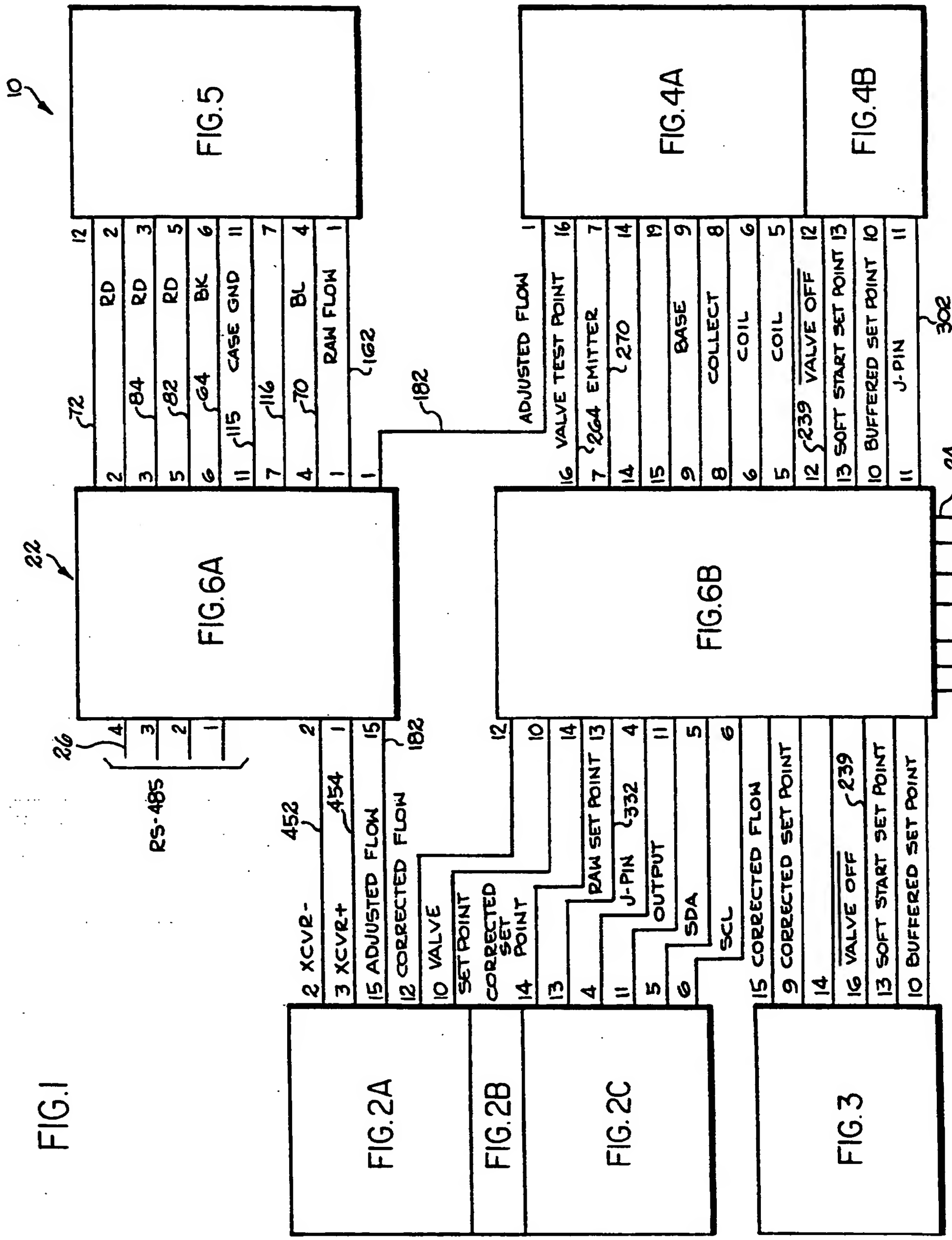
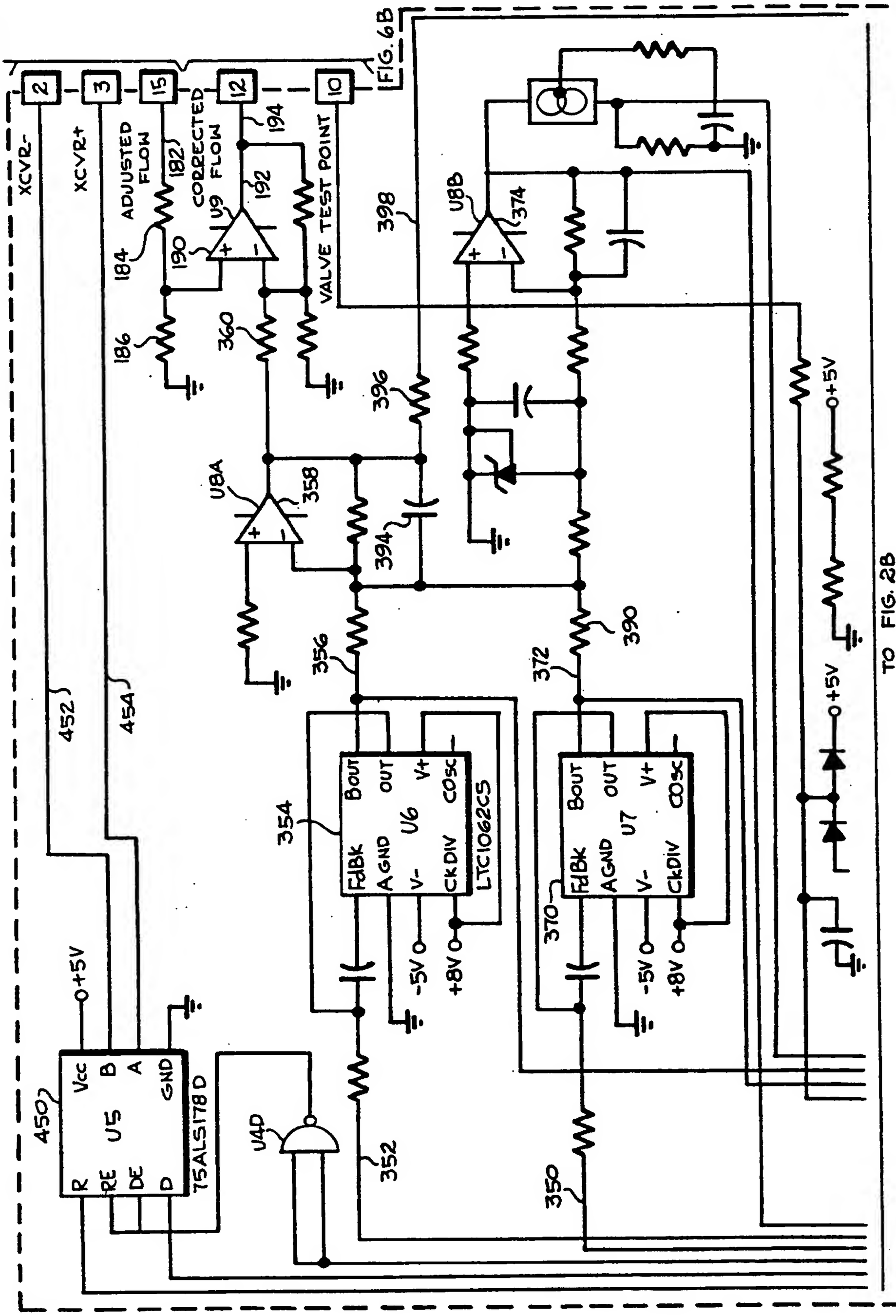


FIG.1

NSDOCID: <WO_____9325950A1_I_>

FIG. 2A



3NSDOCID: <WO_____9325950A1_I_>

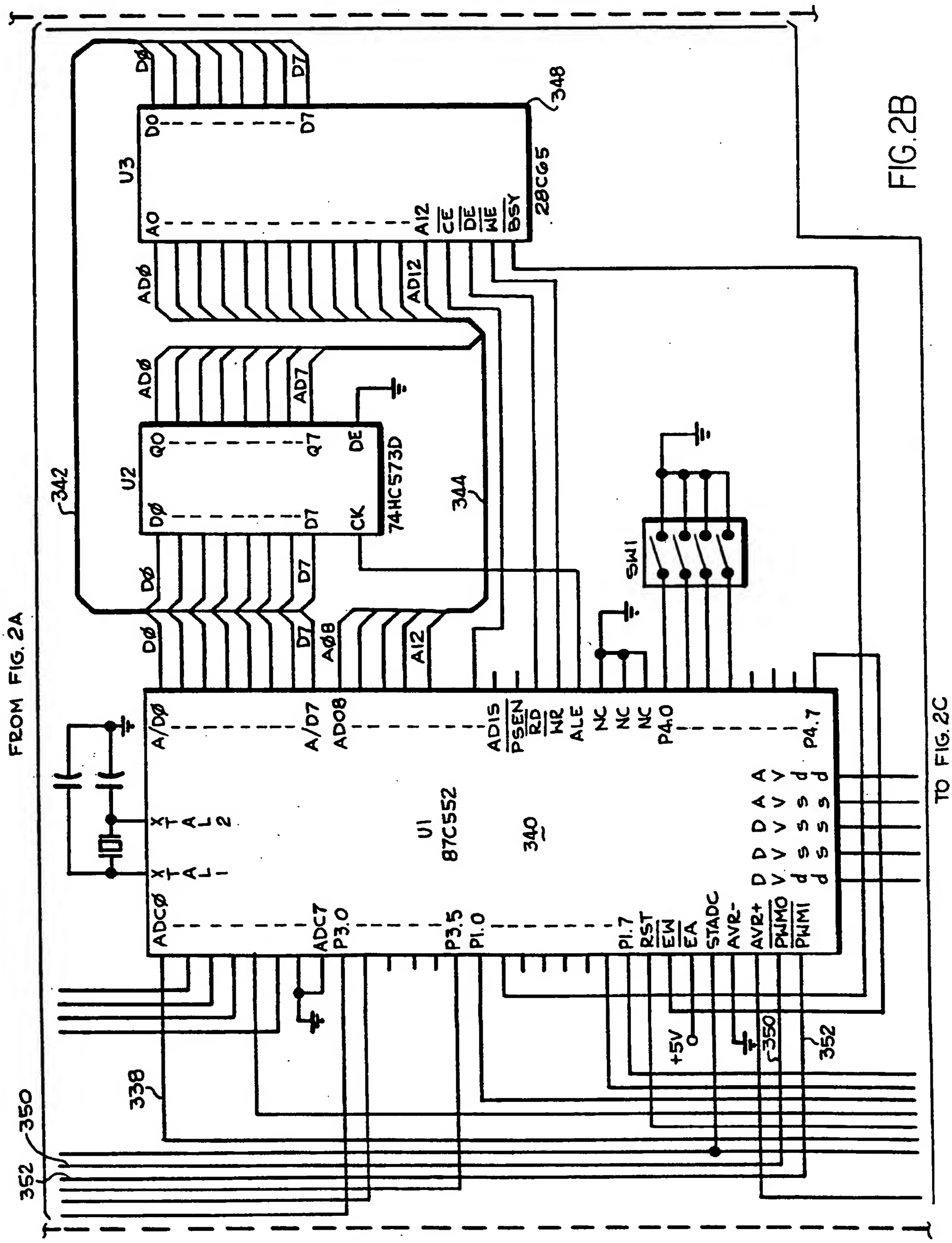


FIG. 2B

NSDOCID: <WO_____9325950A1_I_>

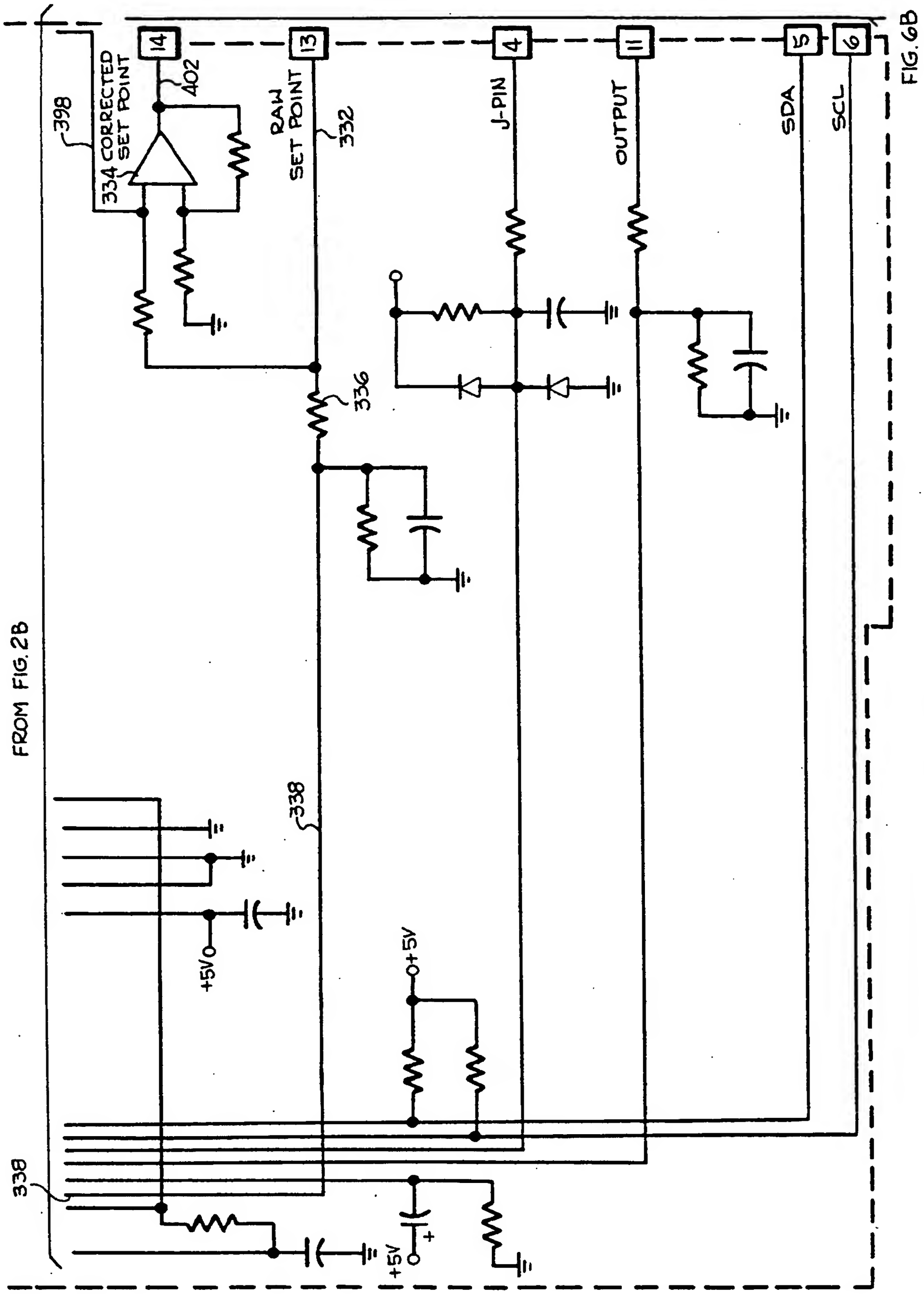
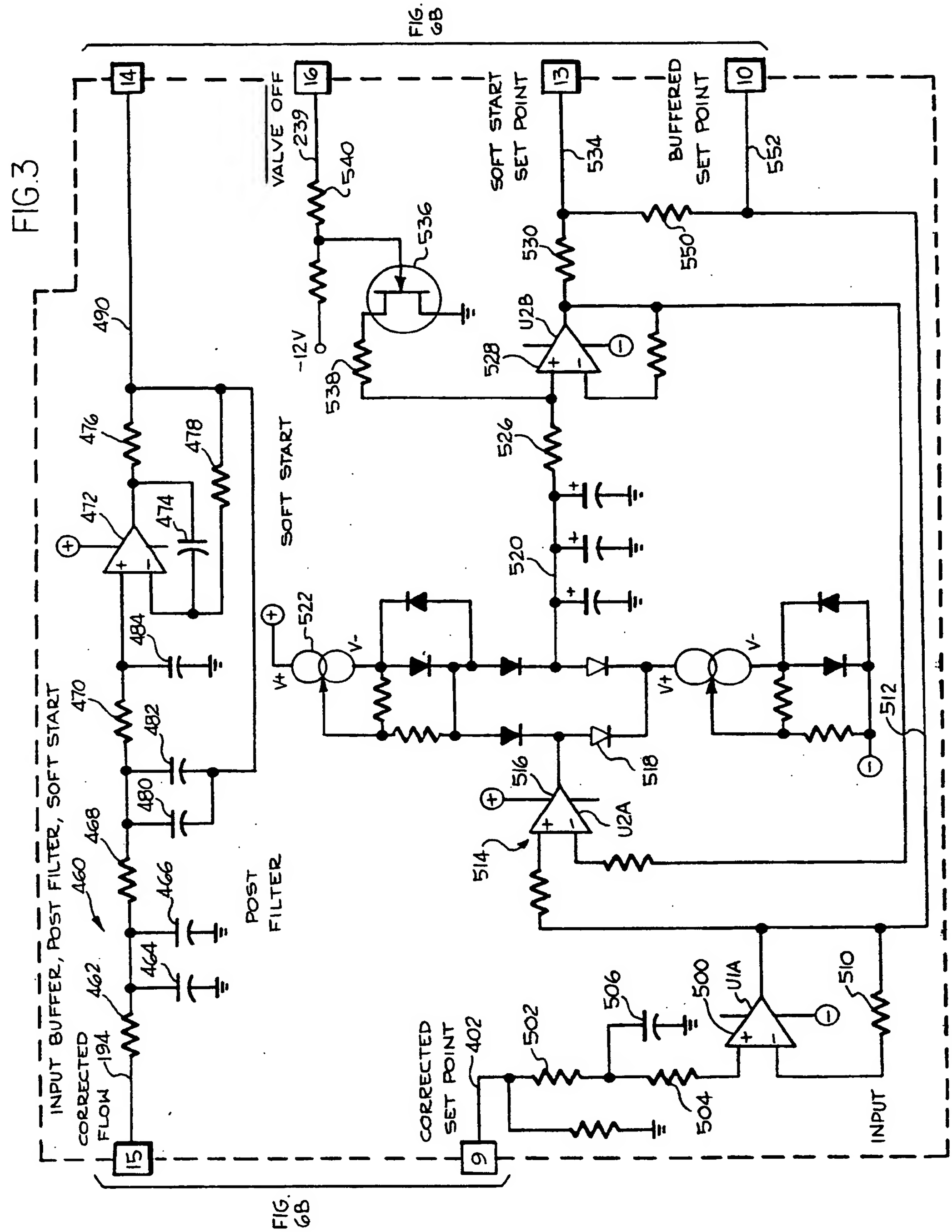


FIG. 2C

3NSDOCID: <WO_____9325950A1_I_>

FIG. 3



WSDOCID: <WO_____9325950A1_I_>

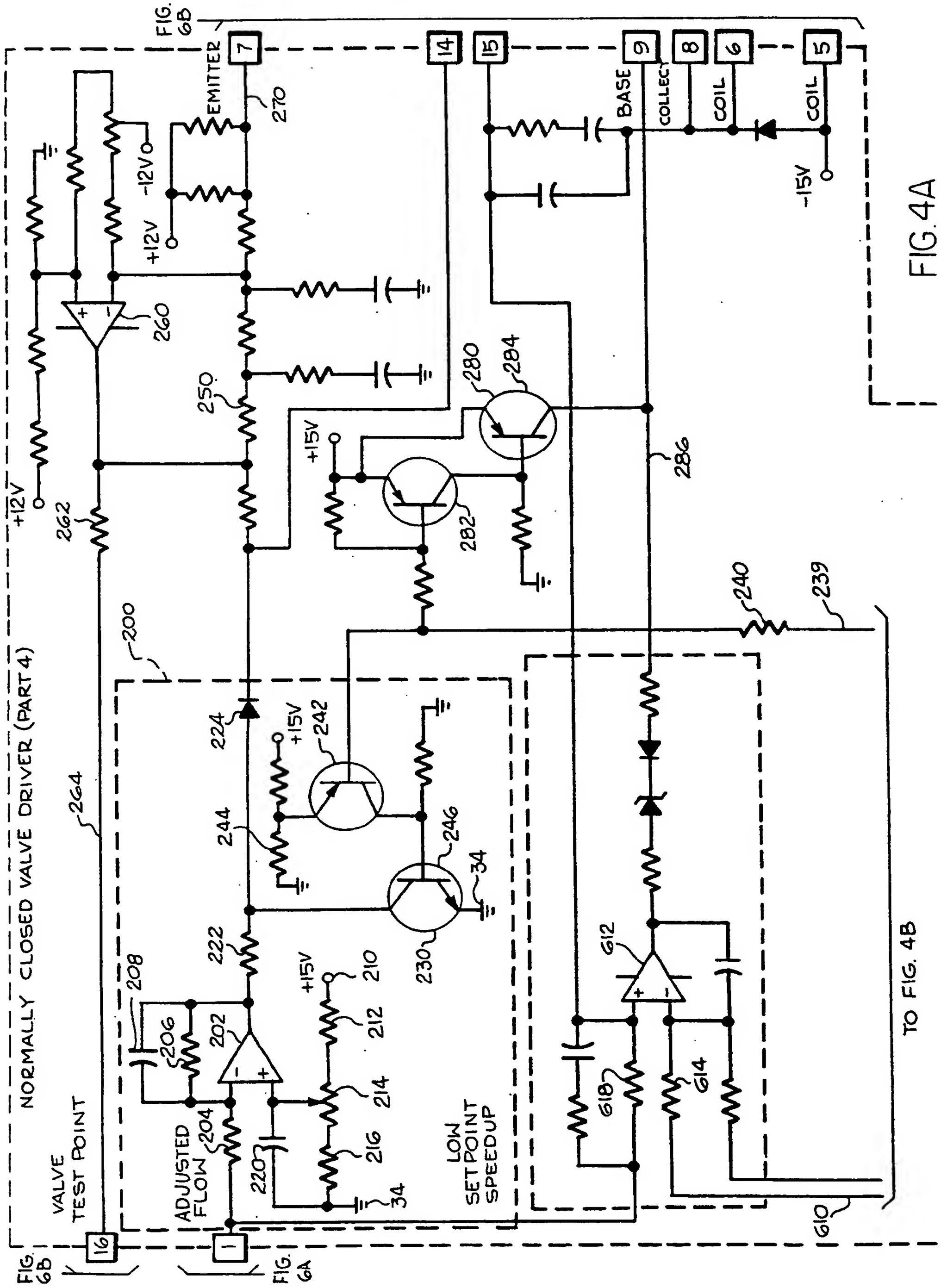


FIG. 4A

3NSDOCID: <WO_____9325950A1_I_>

FIG.4B

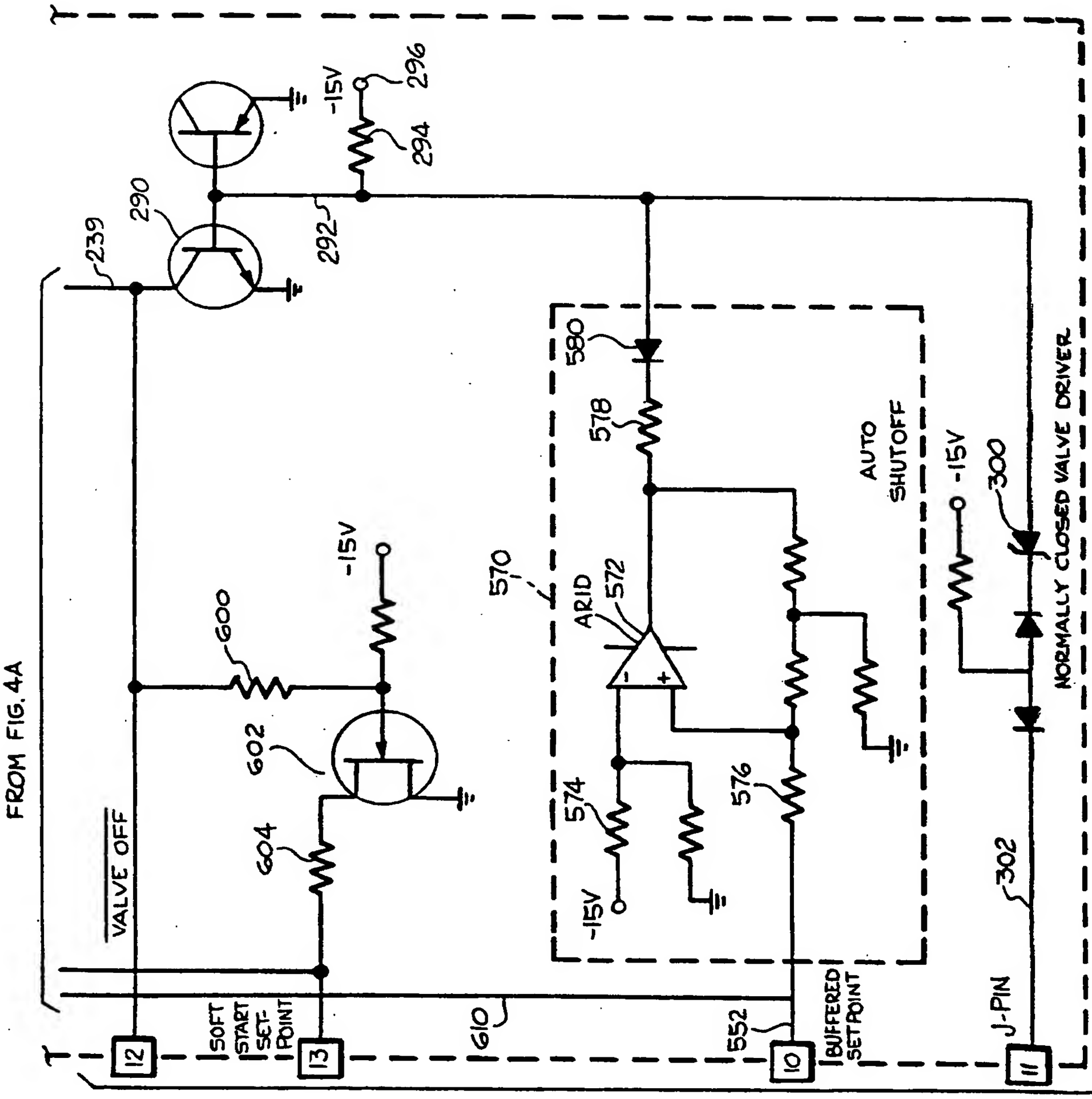
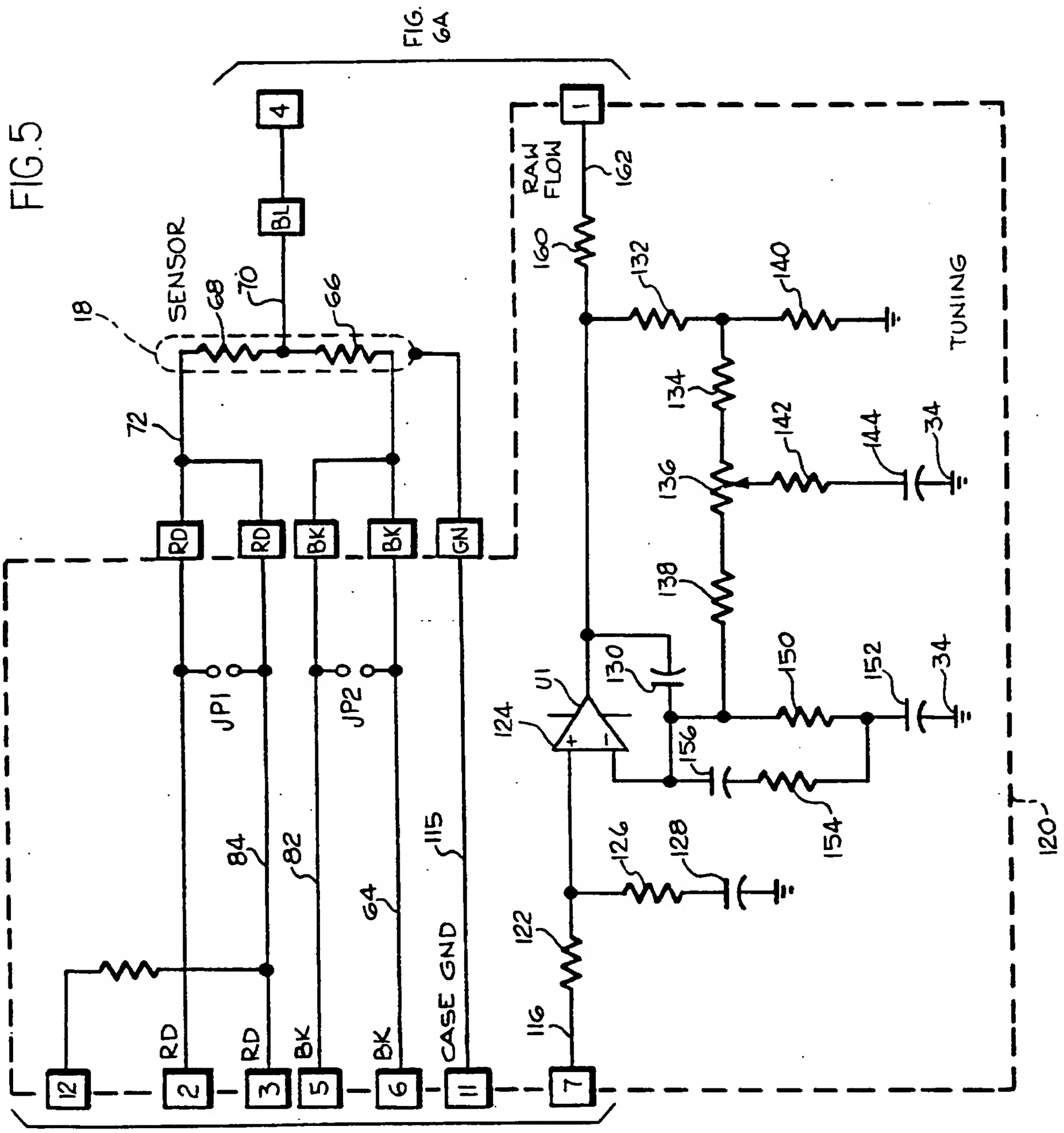


FIG.6B

SDOCID: <WO_____9325950A1_I_>

FIG.5



SUBSTITUTE SHEET

3NSDOCID: <WO_____9325950A1_I_>

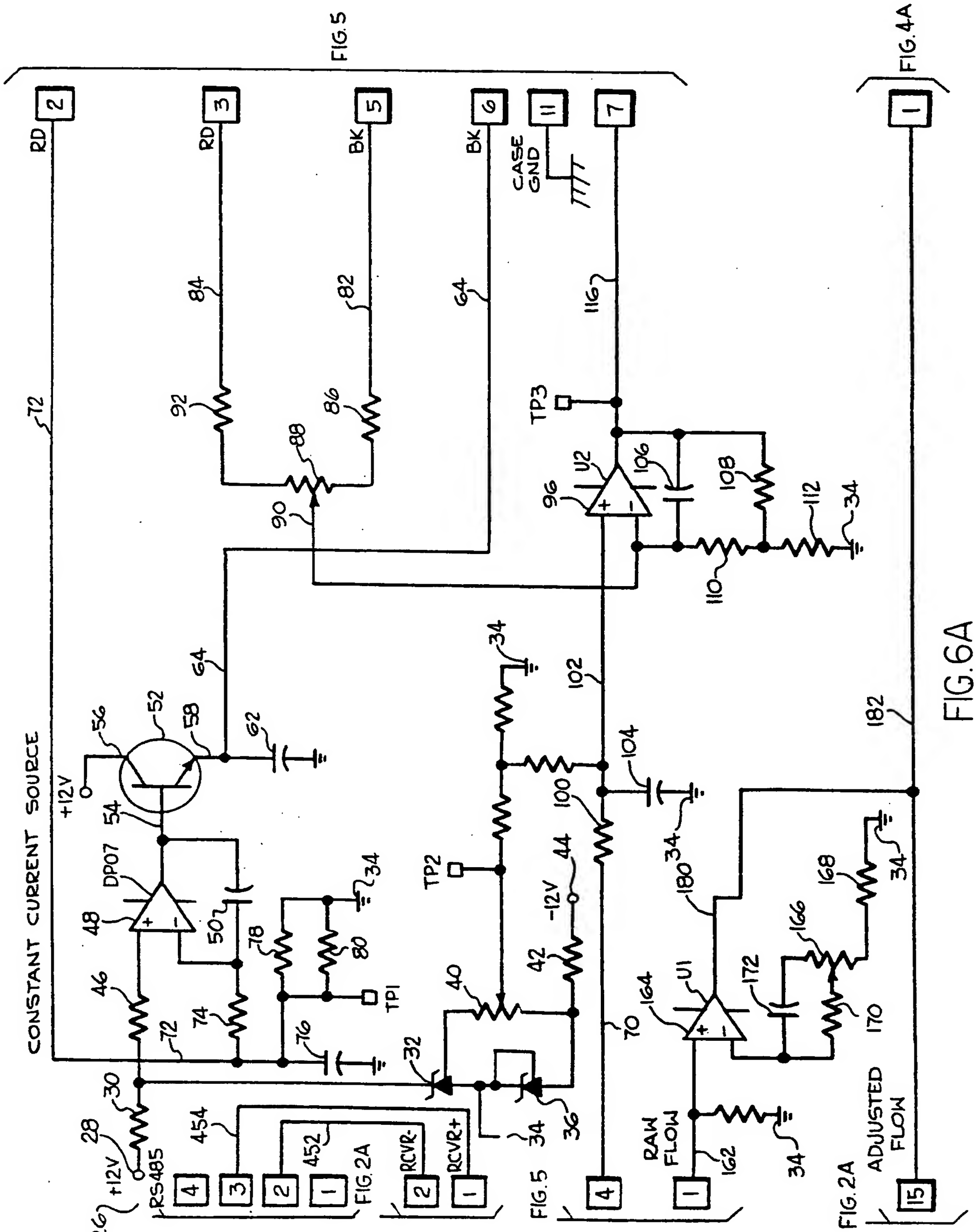


FIG.6A

NSDOCID: <WO_____9325950A1_I_>

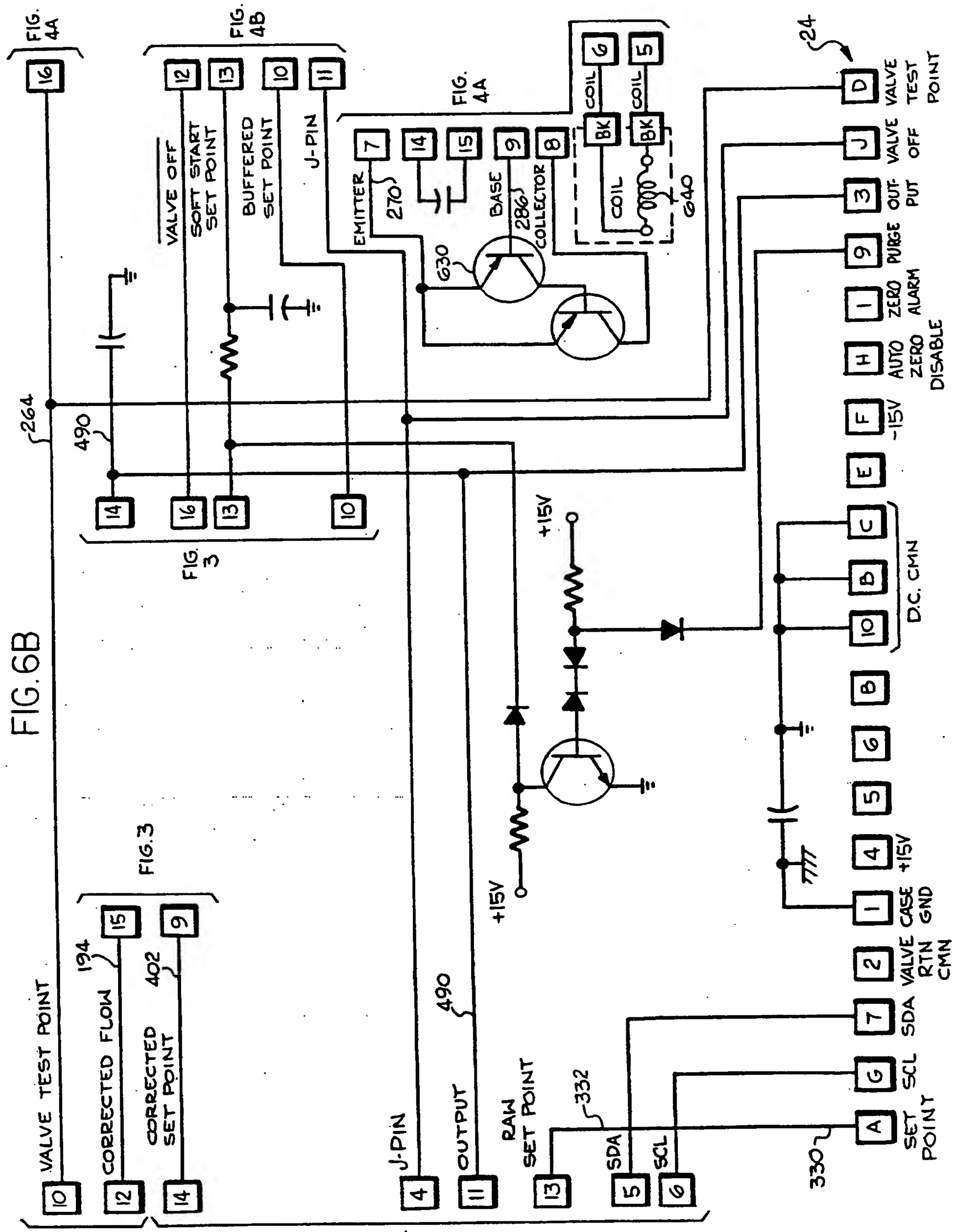
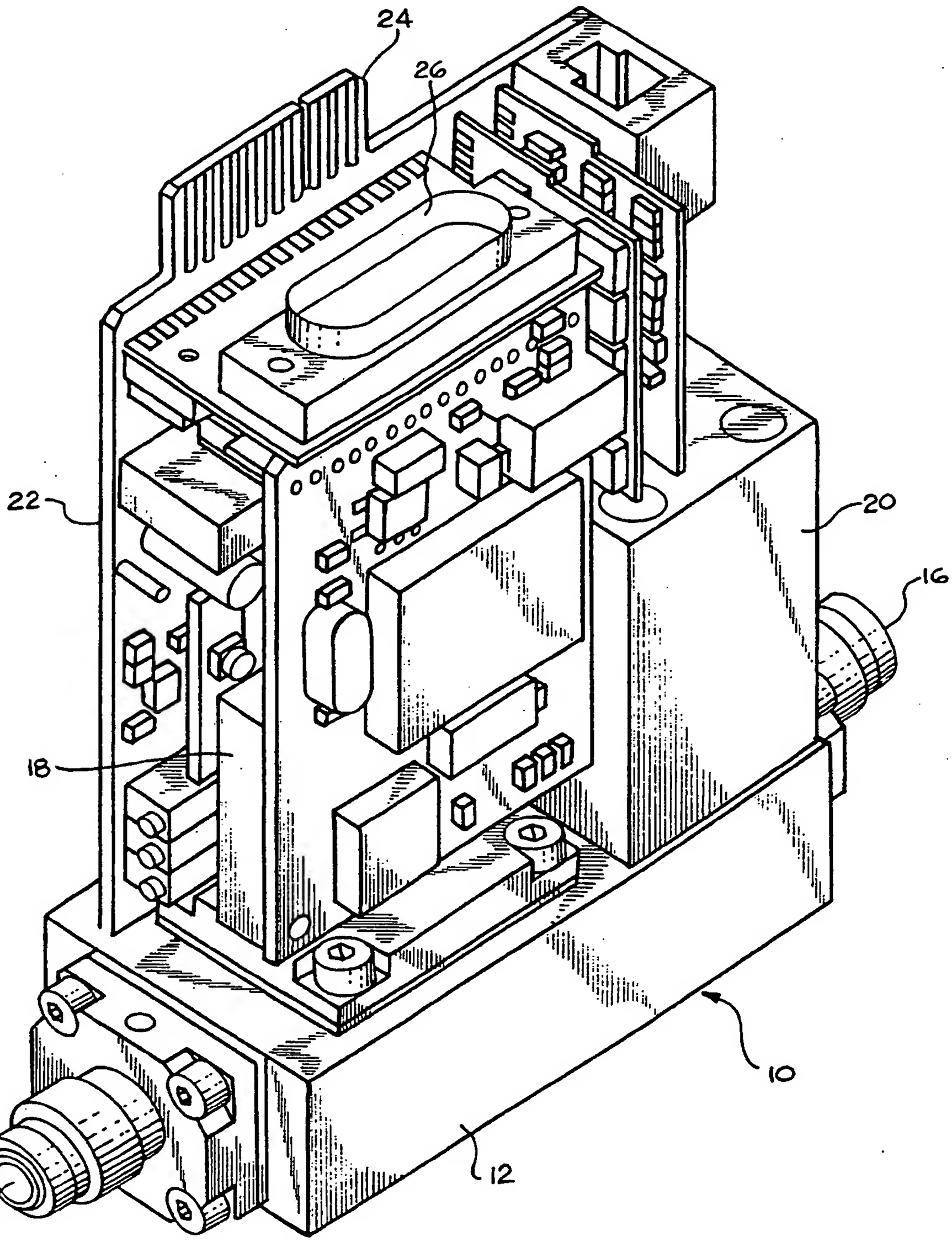


FIG.6B

3NSDOCID: <WO_____9325950A1_I_>

FIG. 7



NSDOCID: <WO_____9325950A1_I_>

12/49

FIG.8

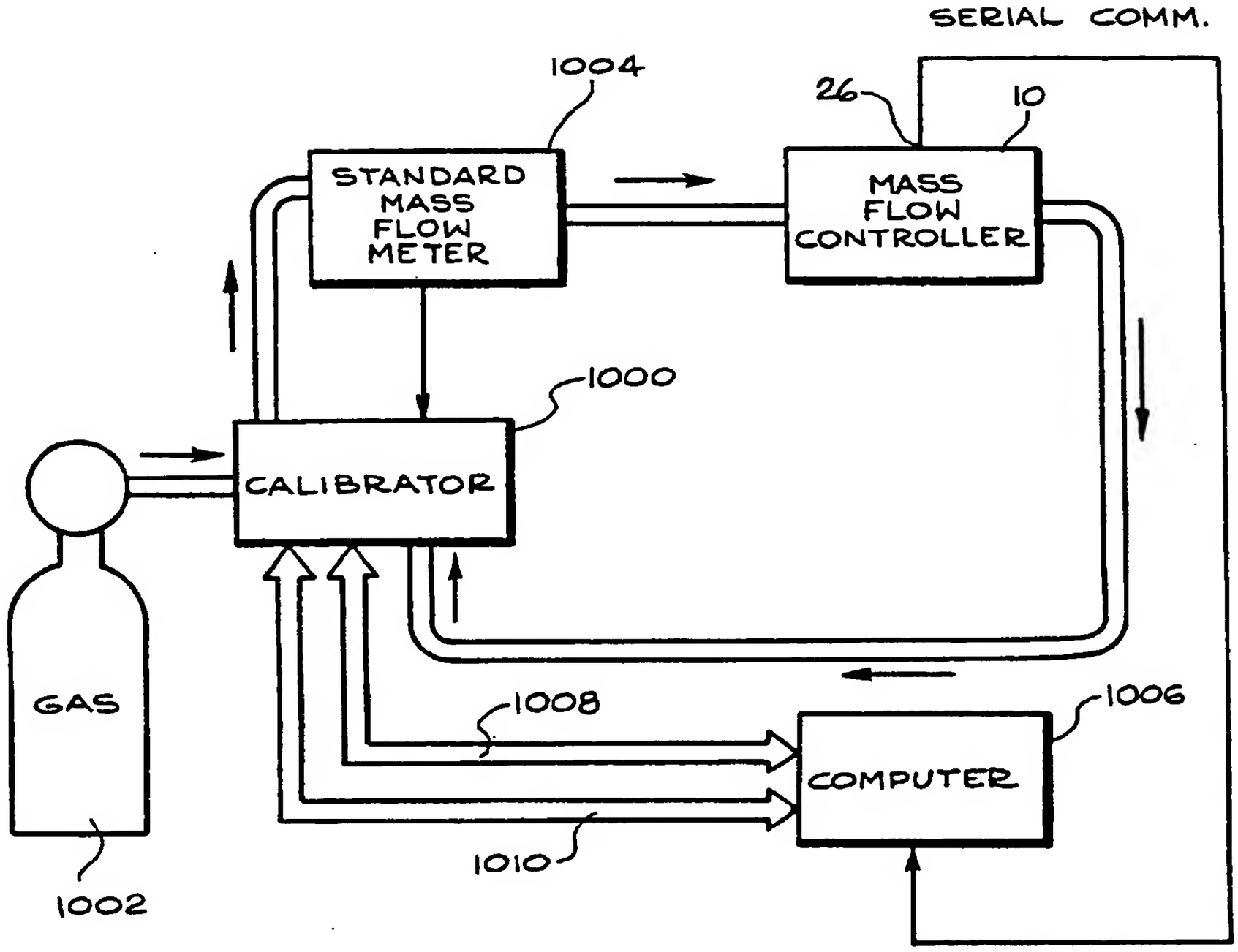
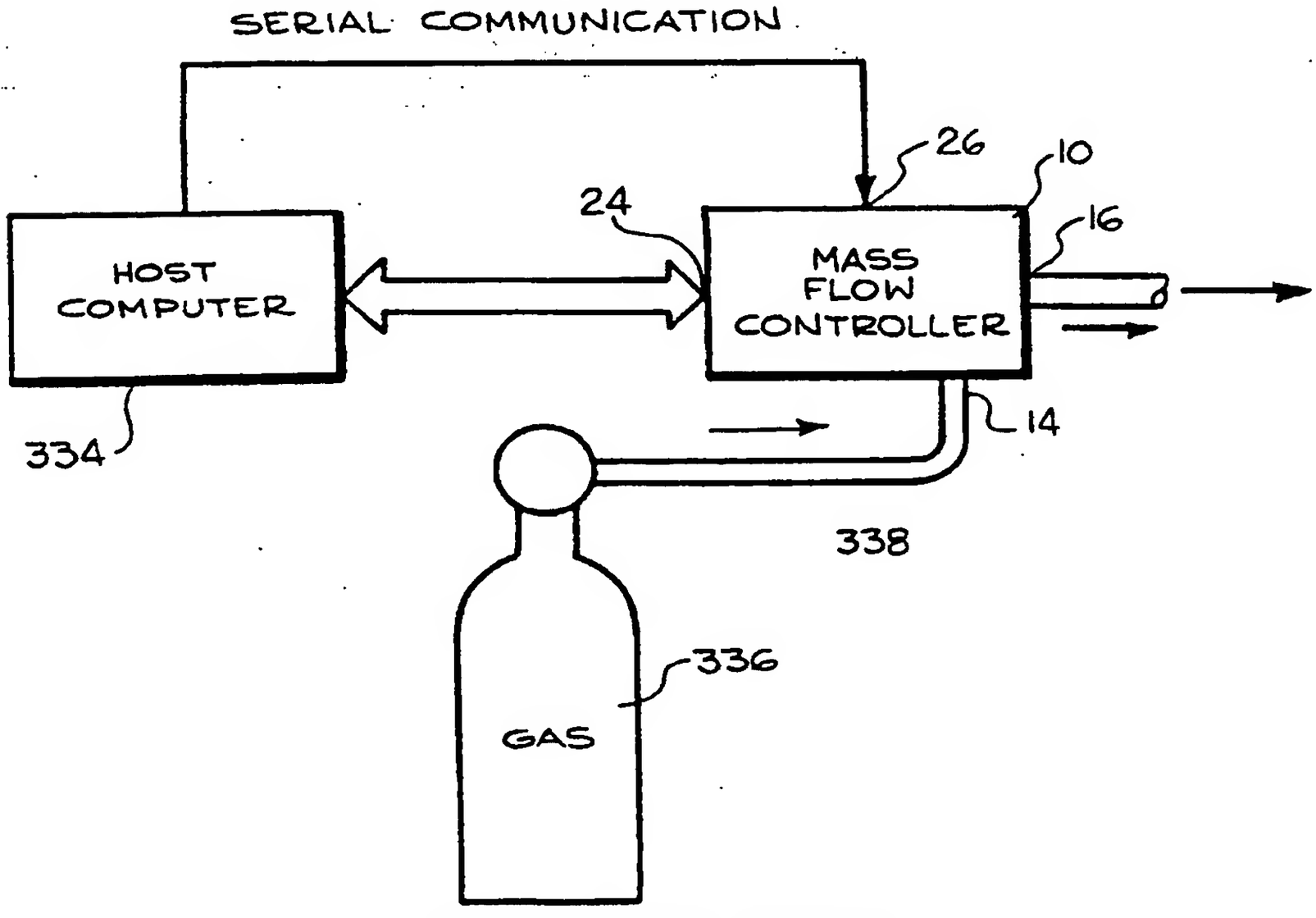
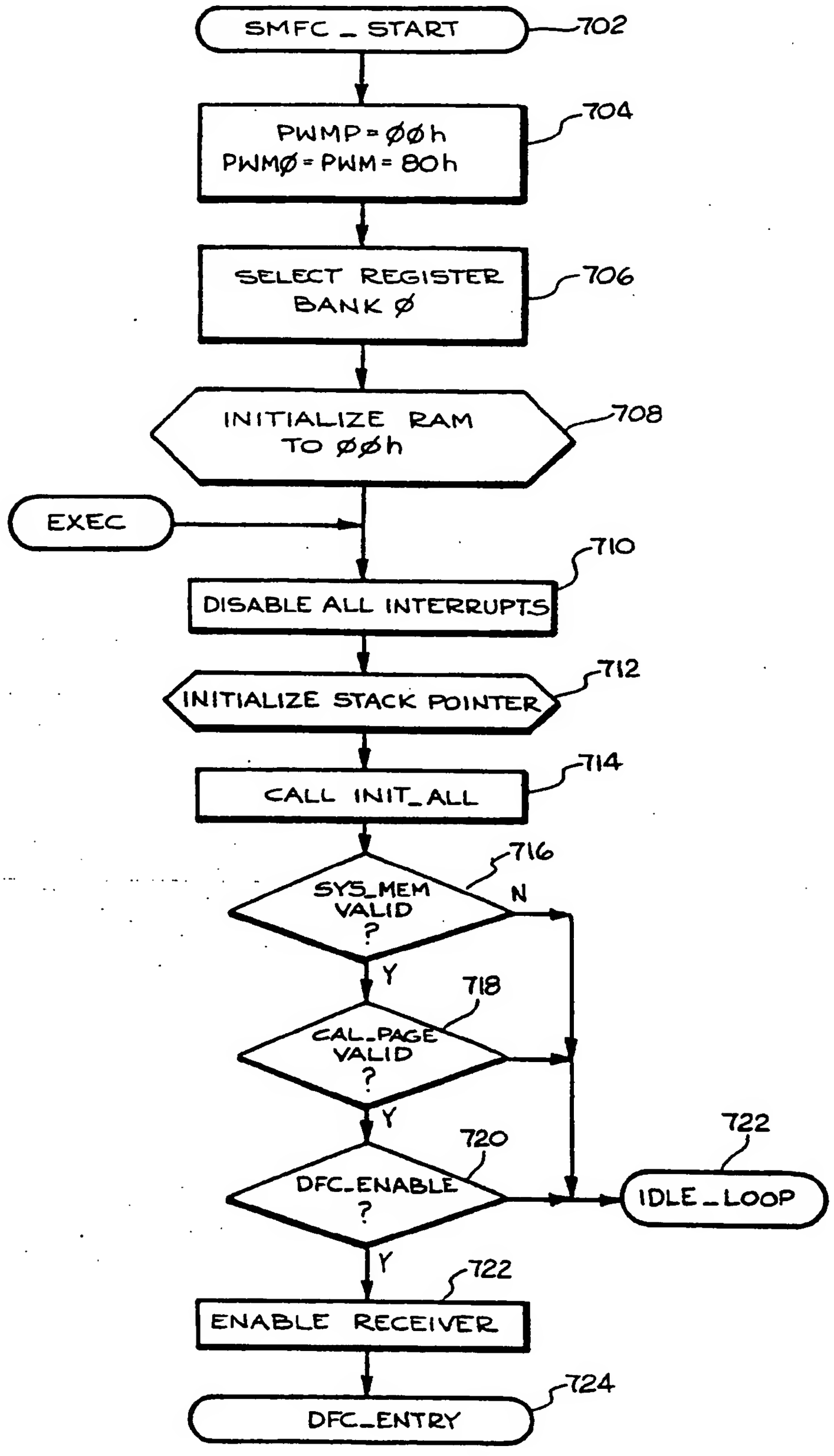


FIG.9



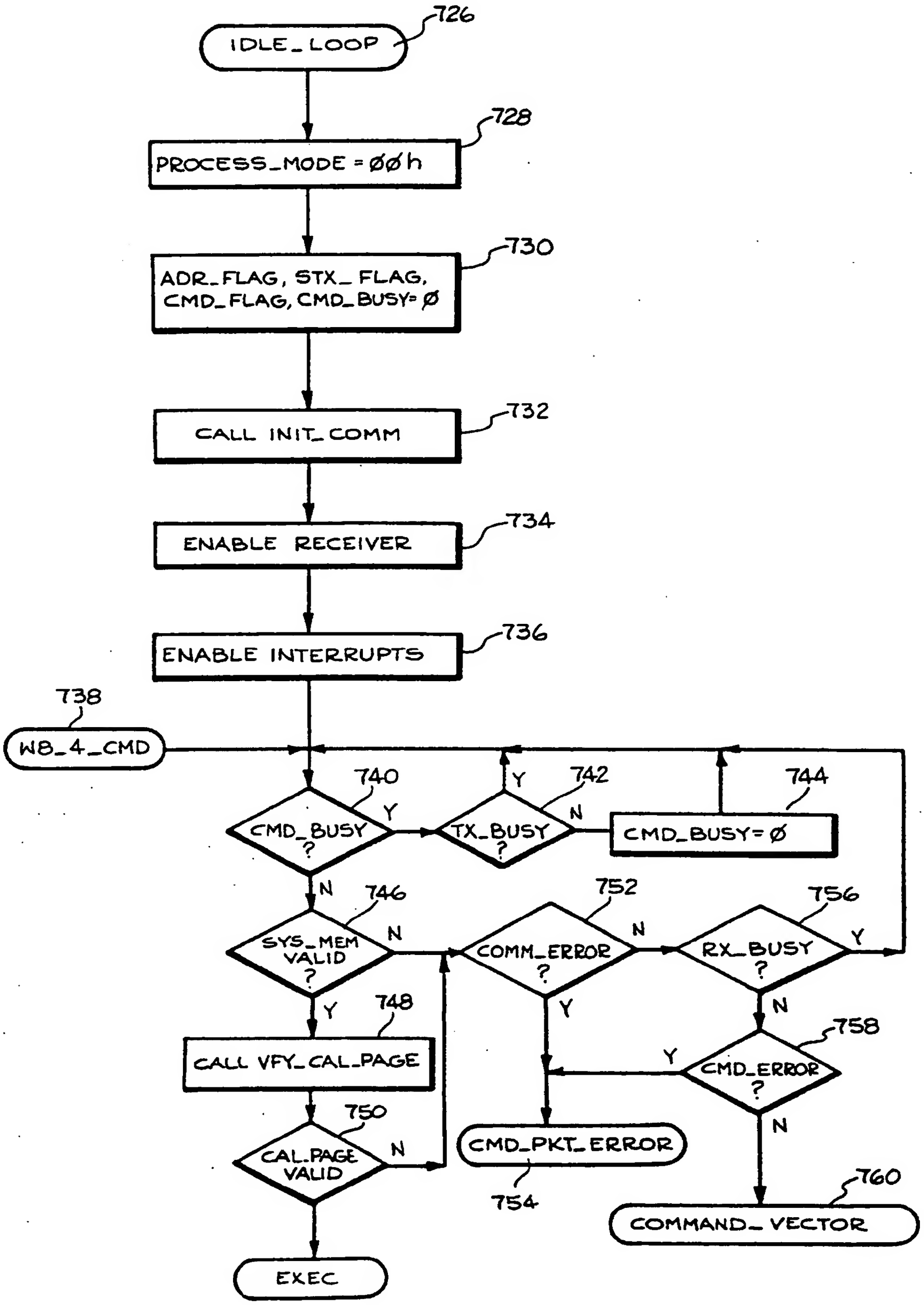
3NSDOCID: <WO_____9325950A1_I_>

FIG. 10



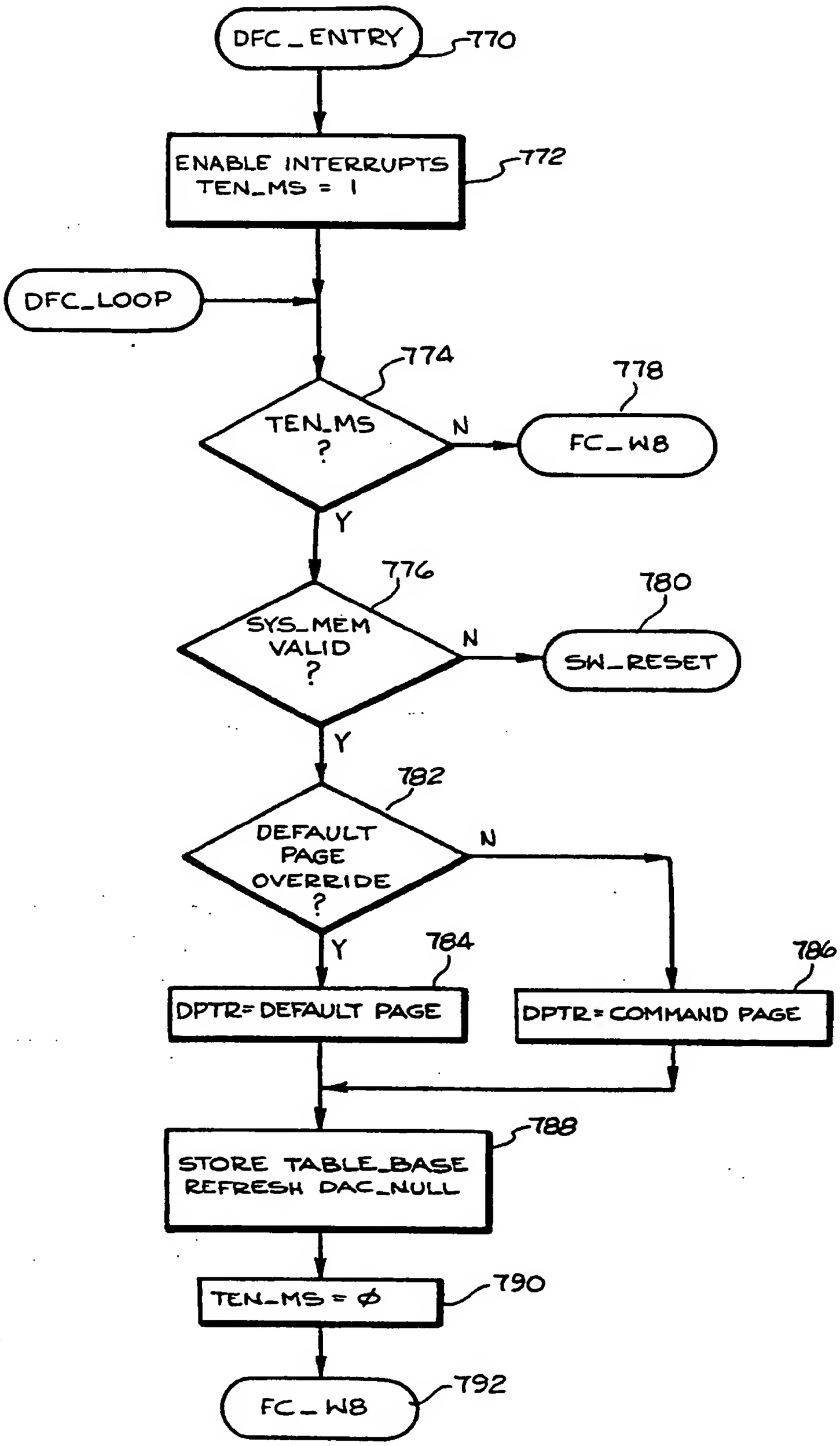
ISDOCID: <WO_____9325950A1_I_>

FIG. 11



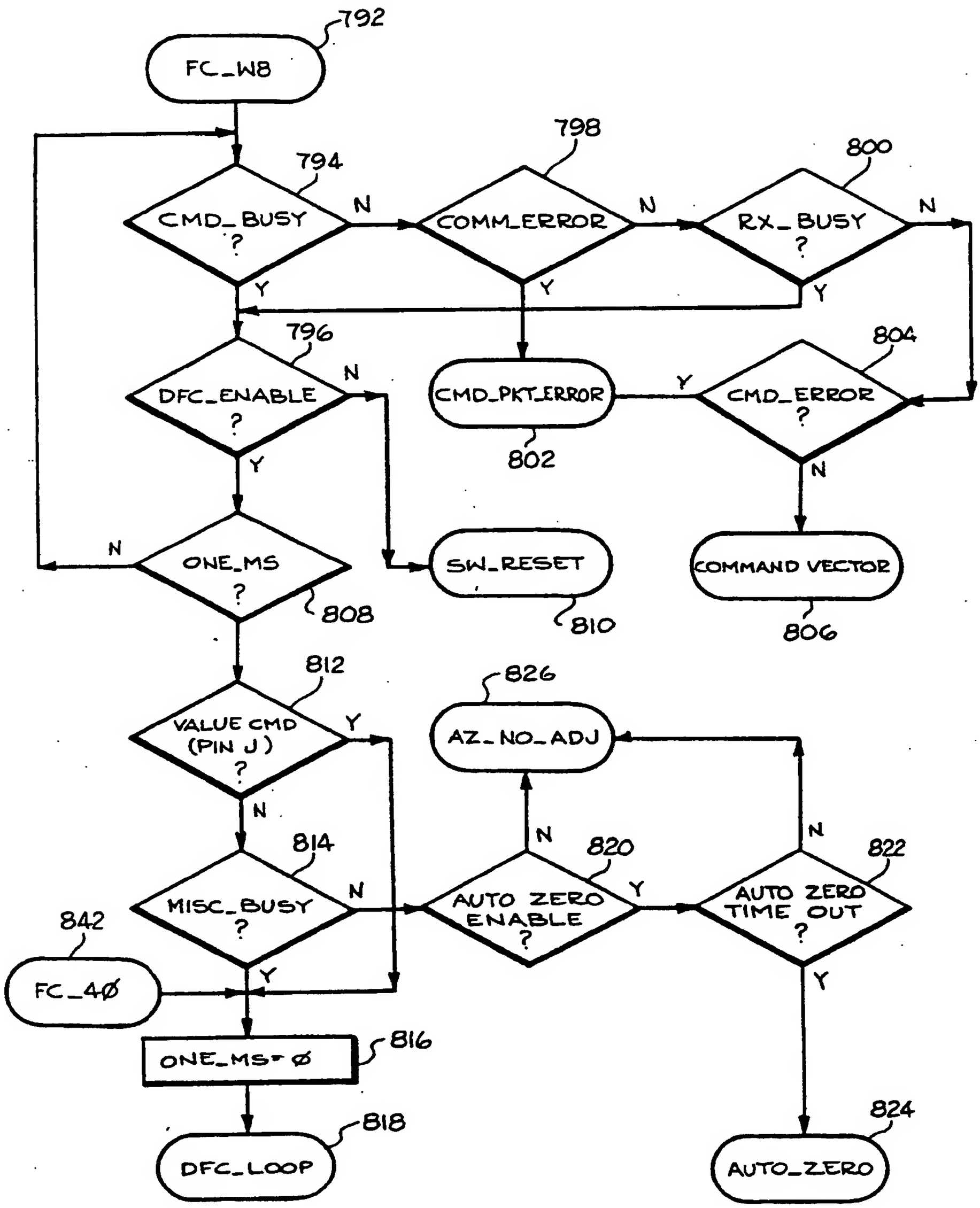
3NSDOCID: <WO_____9325950A1_I_>

FIG. 12A



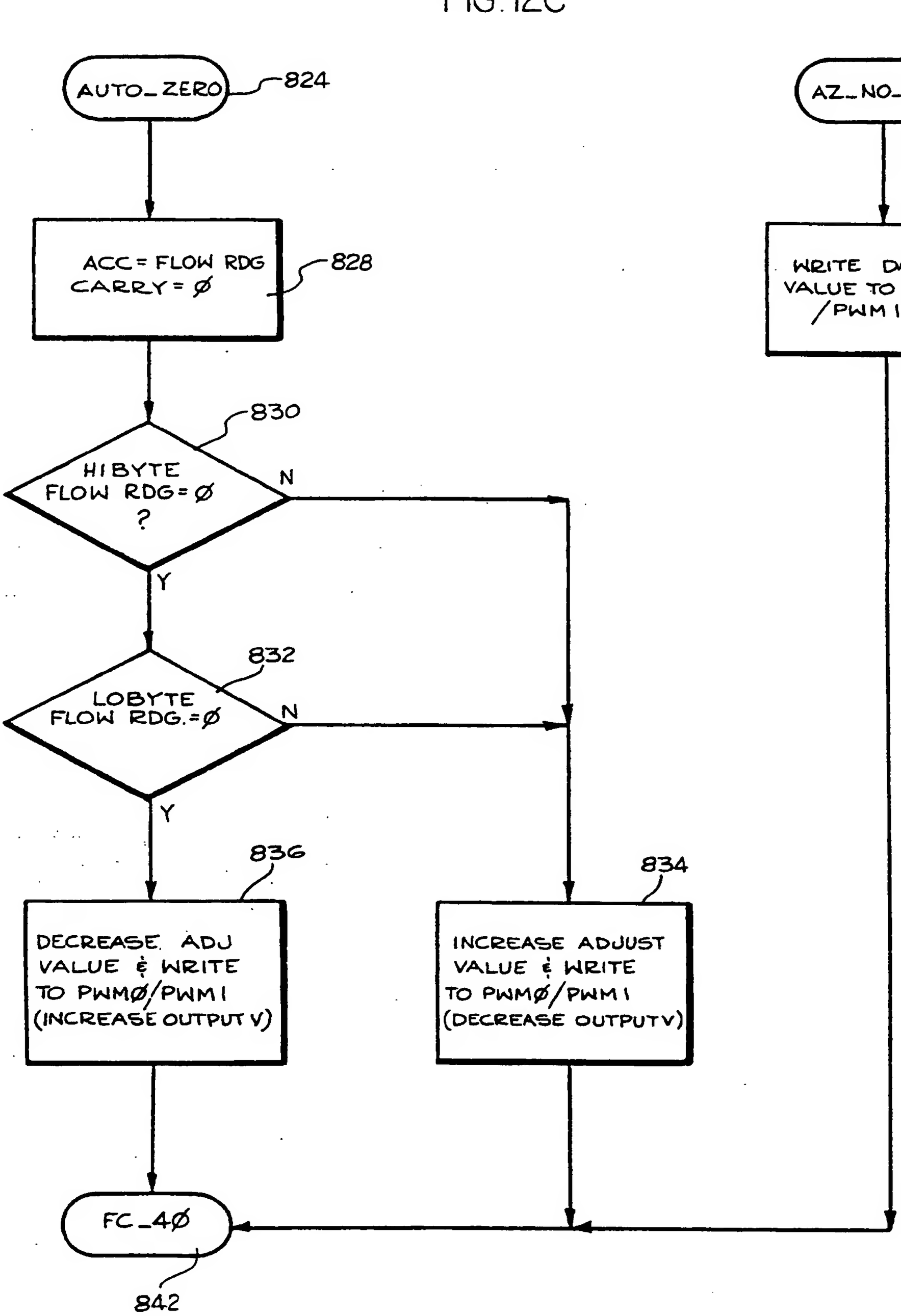
NSDOCID: <WO_____9325950A1_I_>

FIG.12B



NSDOCID: <WO_____9325950A1_I_>

FIG. 12C



ISDOCID: <WO_____9325950A1_I_>

FIG.13A

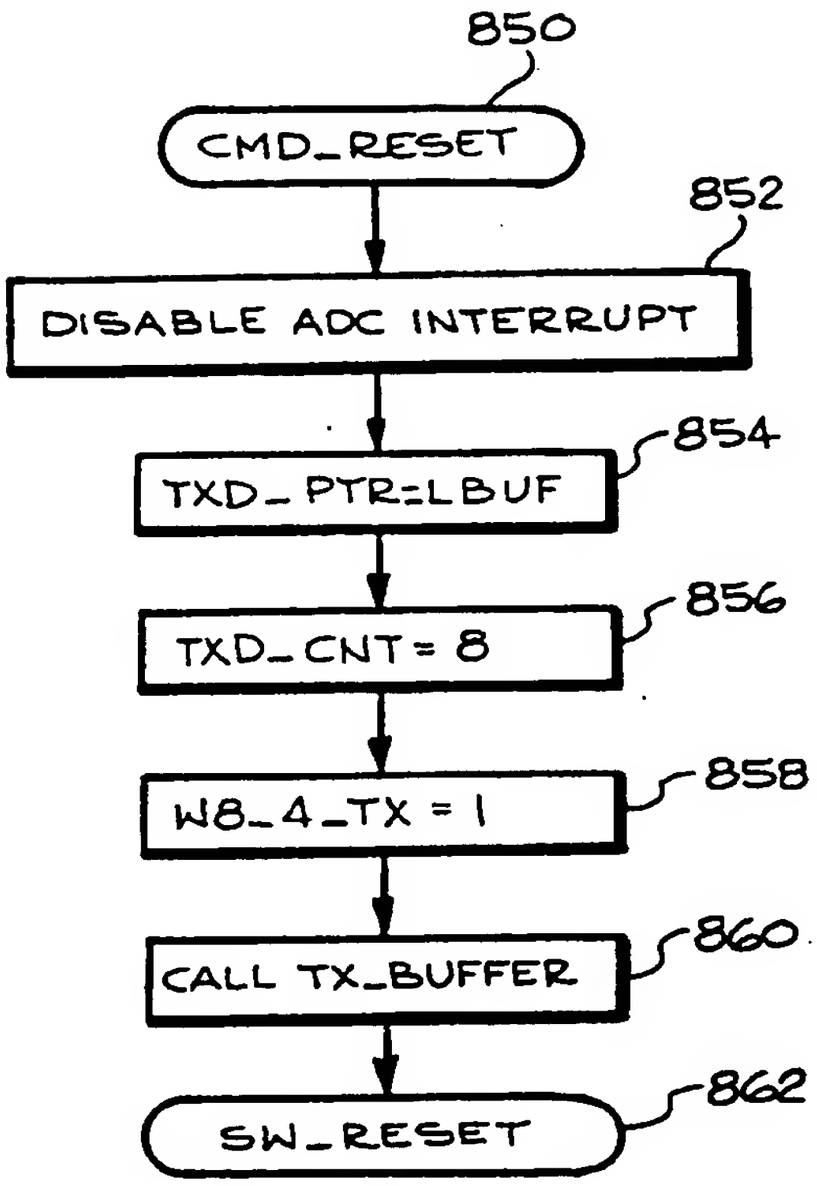
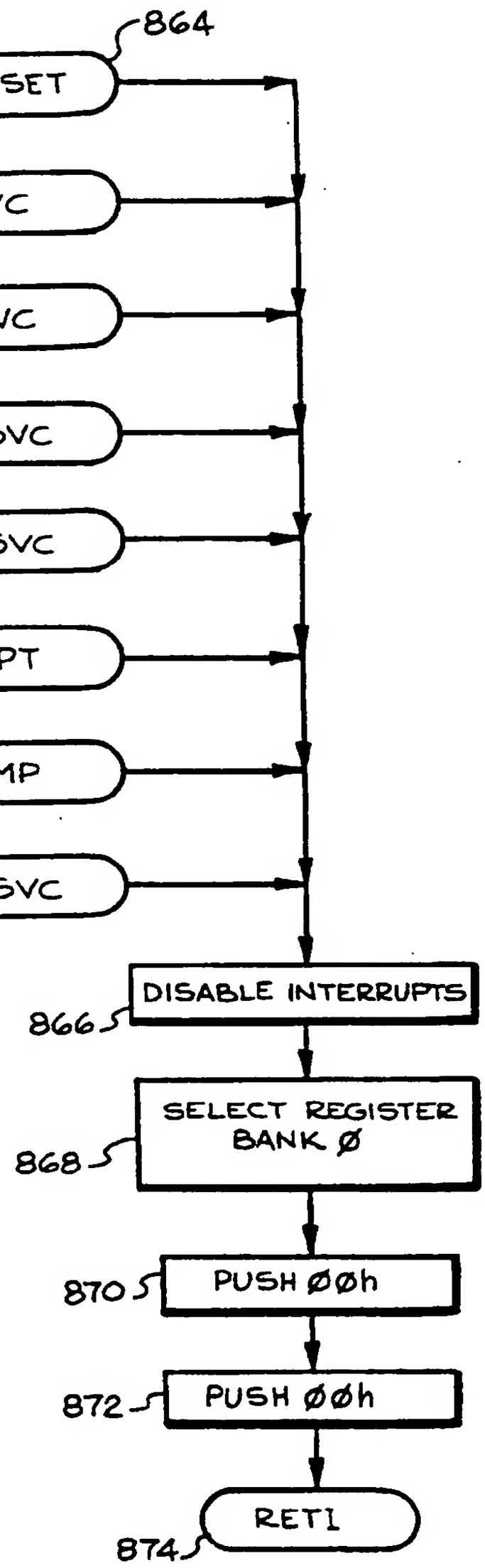
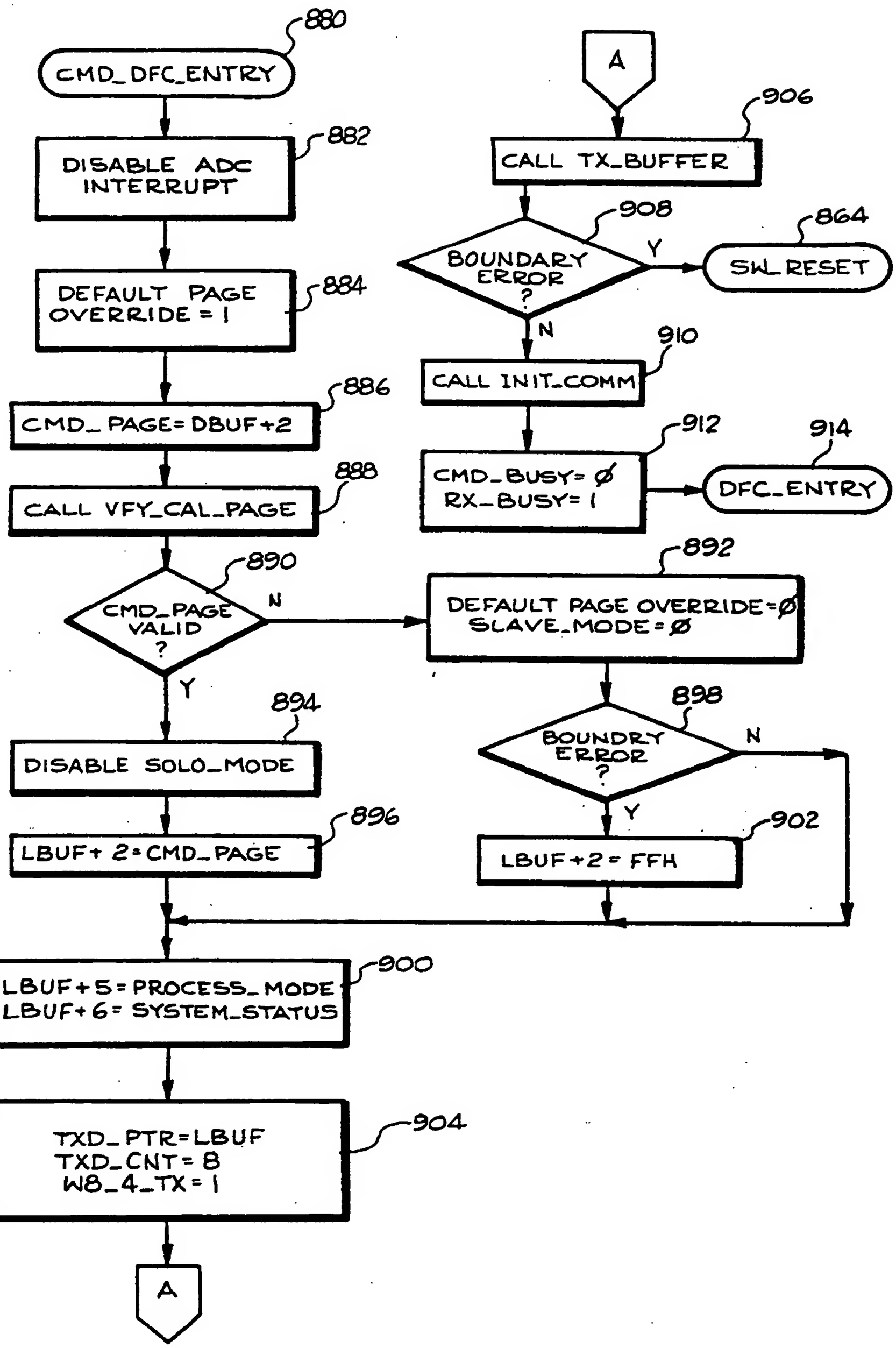


FIG.13B



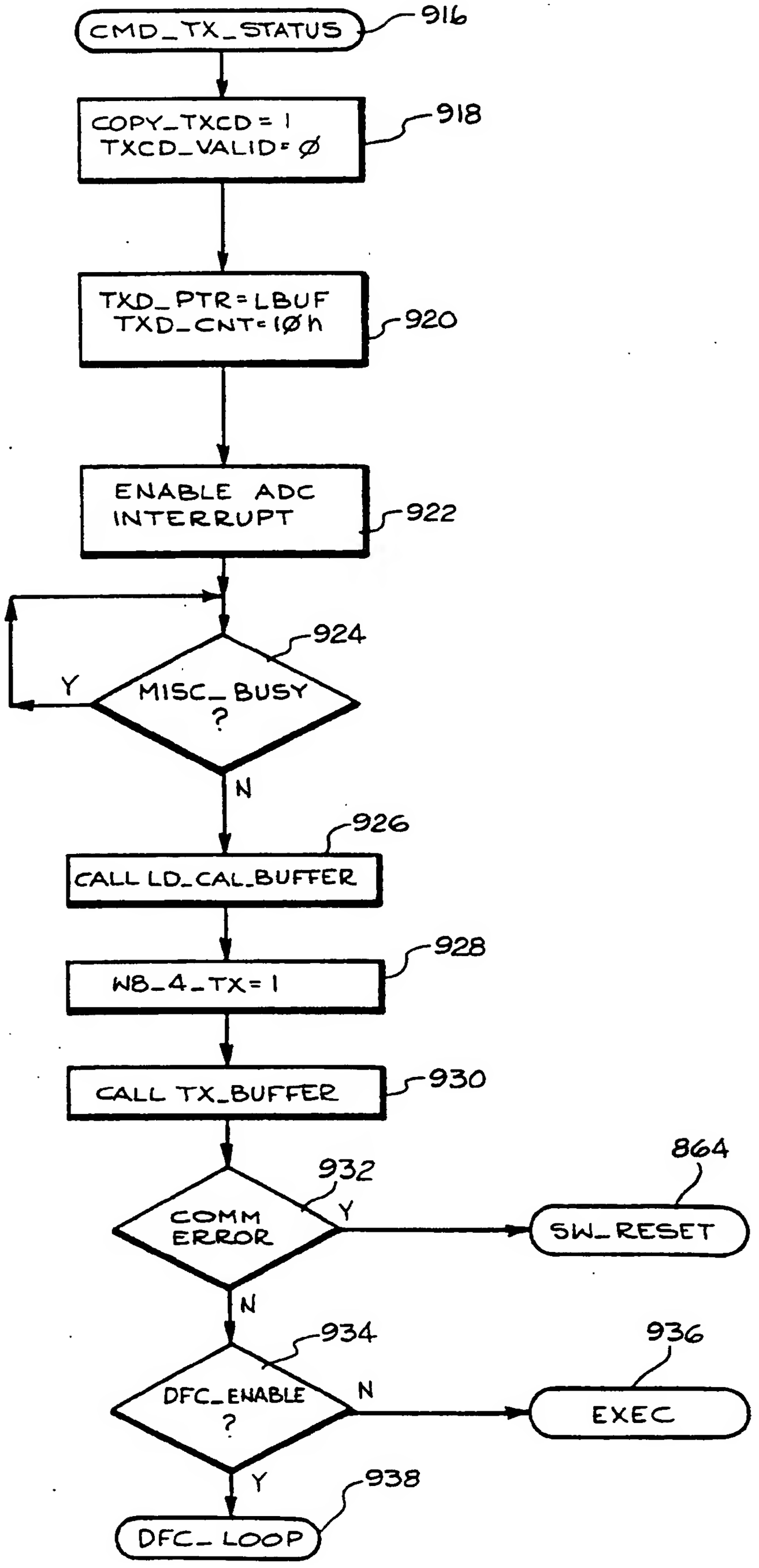
BNSDOCID: <WO_____9325950A1_I_>

FIG.14



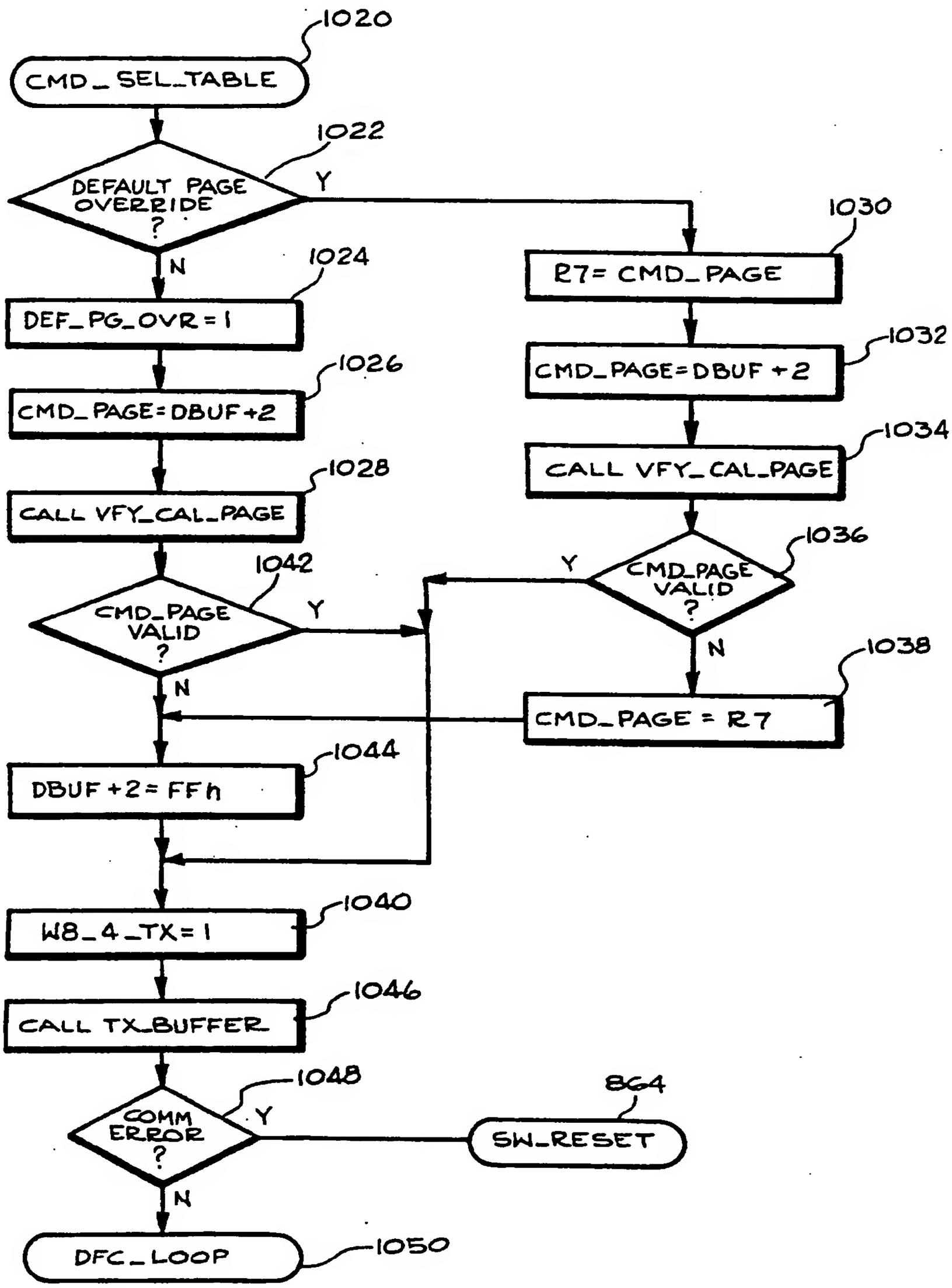
NSDOCID: <WO_____9325950A1_I_>

FIG. 15



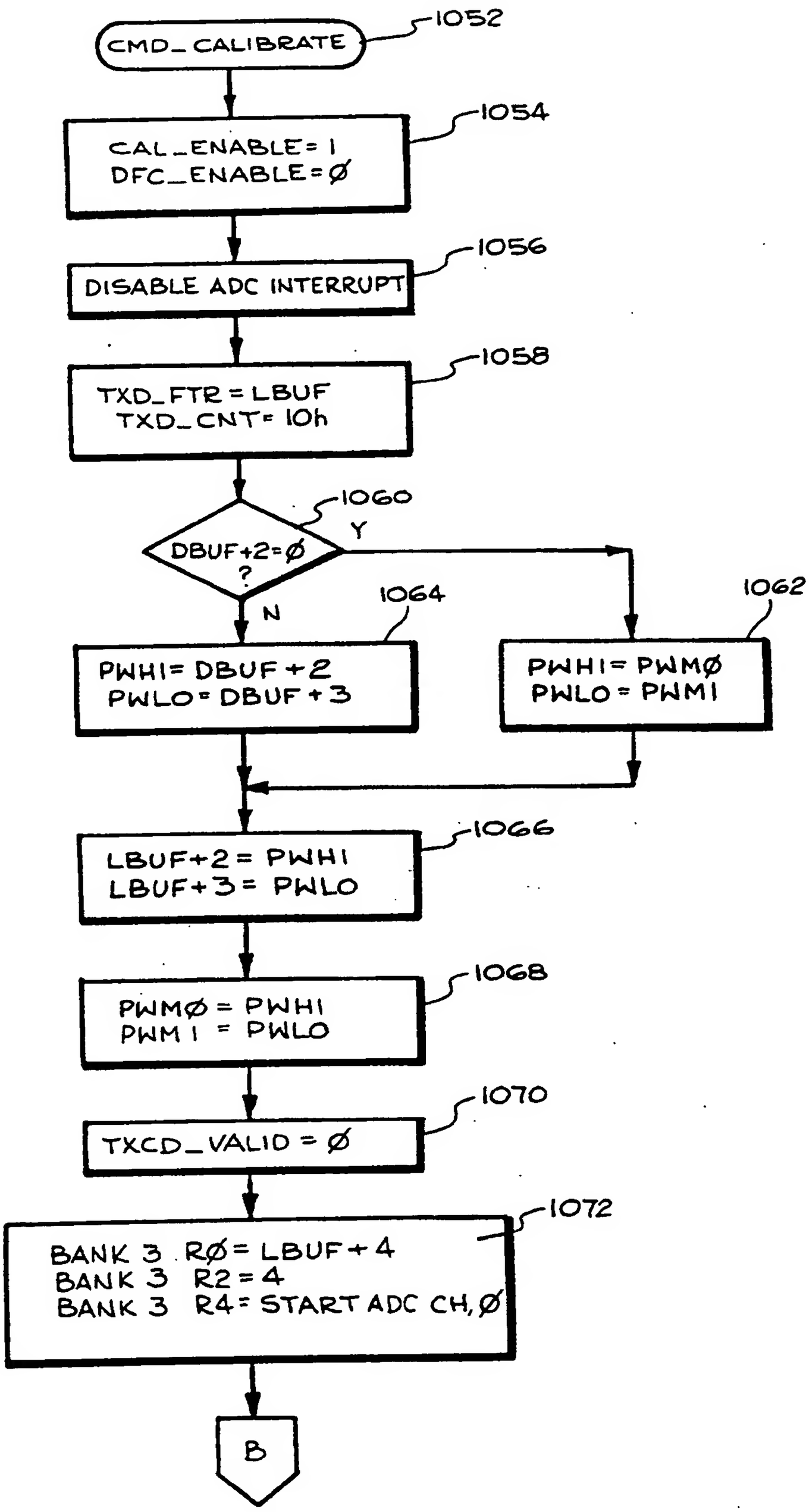
BNSDOCID: <WO_____9325950A1_I_>

FIG.16



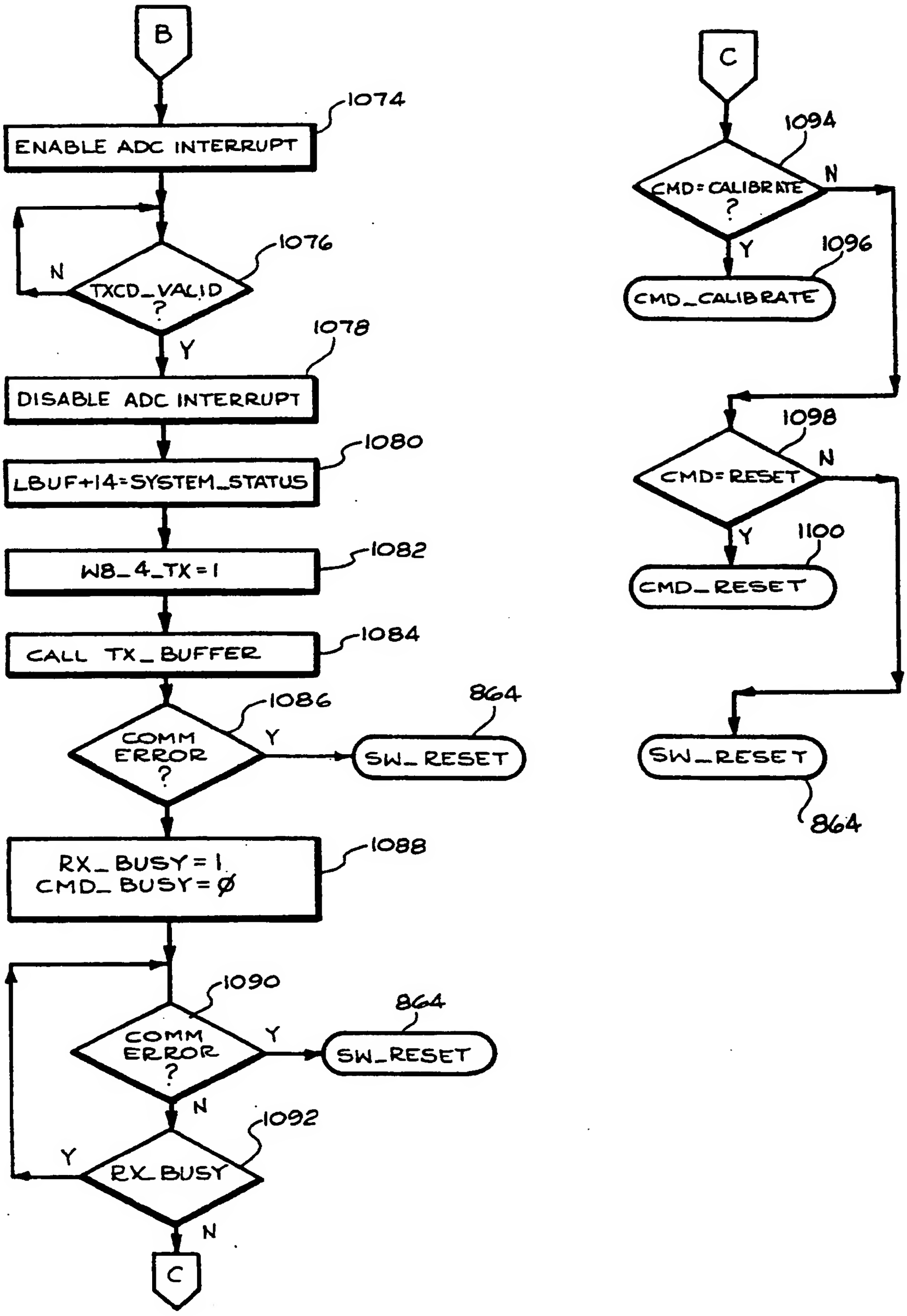
NSDOCID: <WO_____9325950A1_I_>

FIG. 17



3NSDOCID: <WO_____9325950A1_I_>

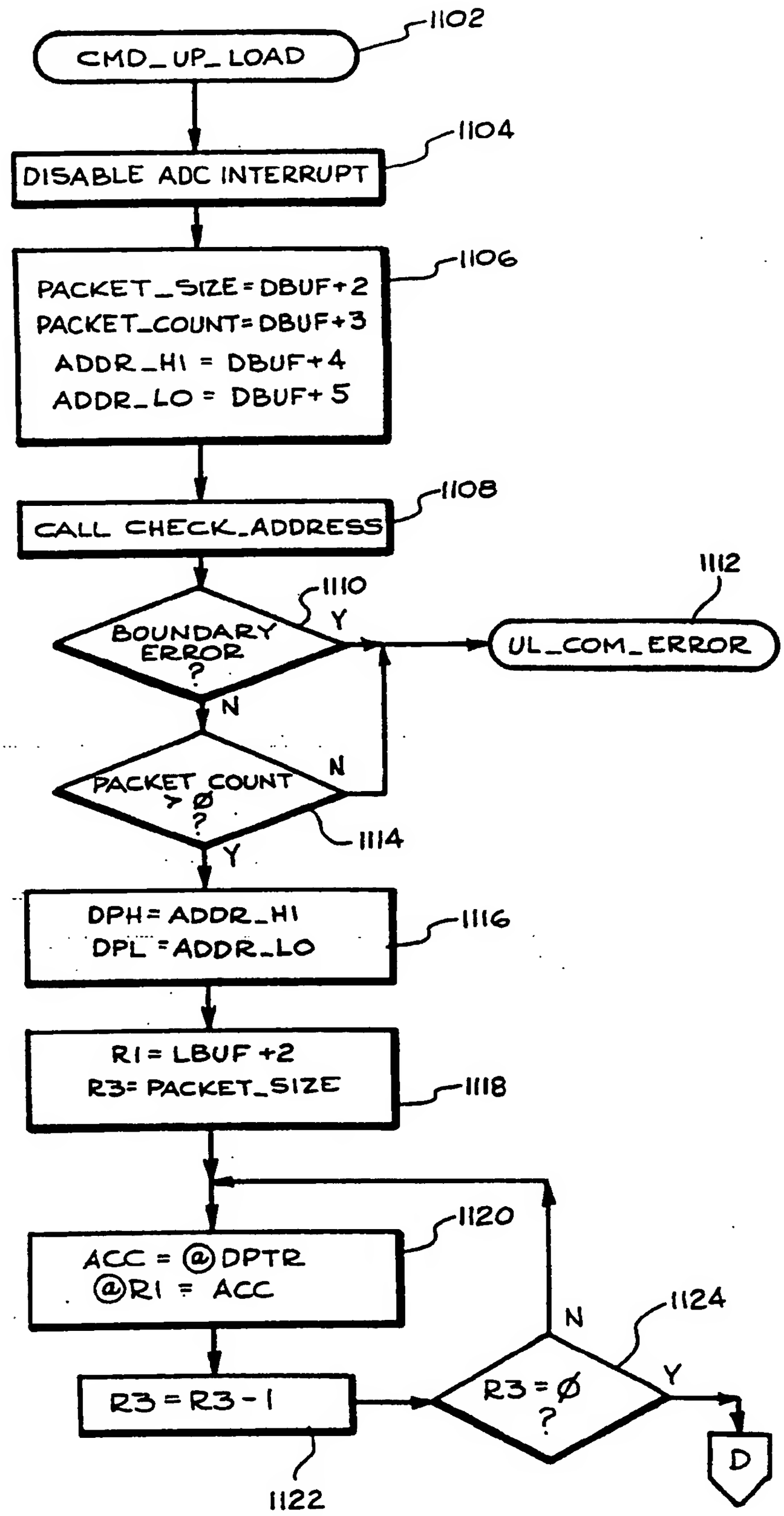
FIG. 18



SUBSTITUTE SHEET

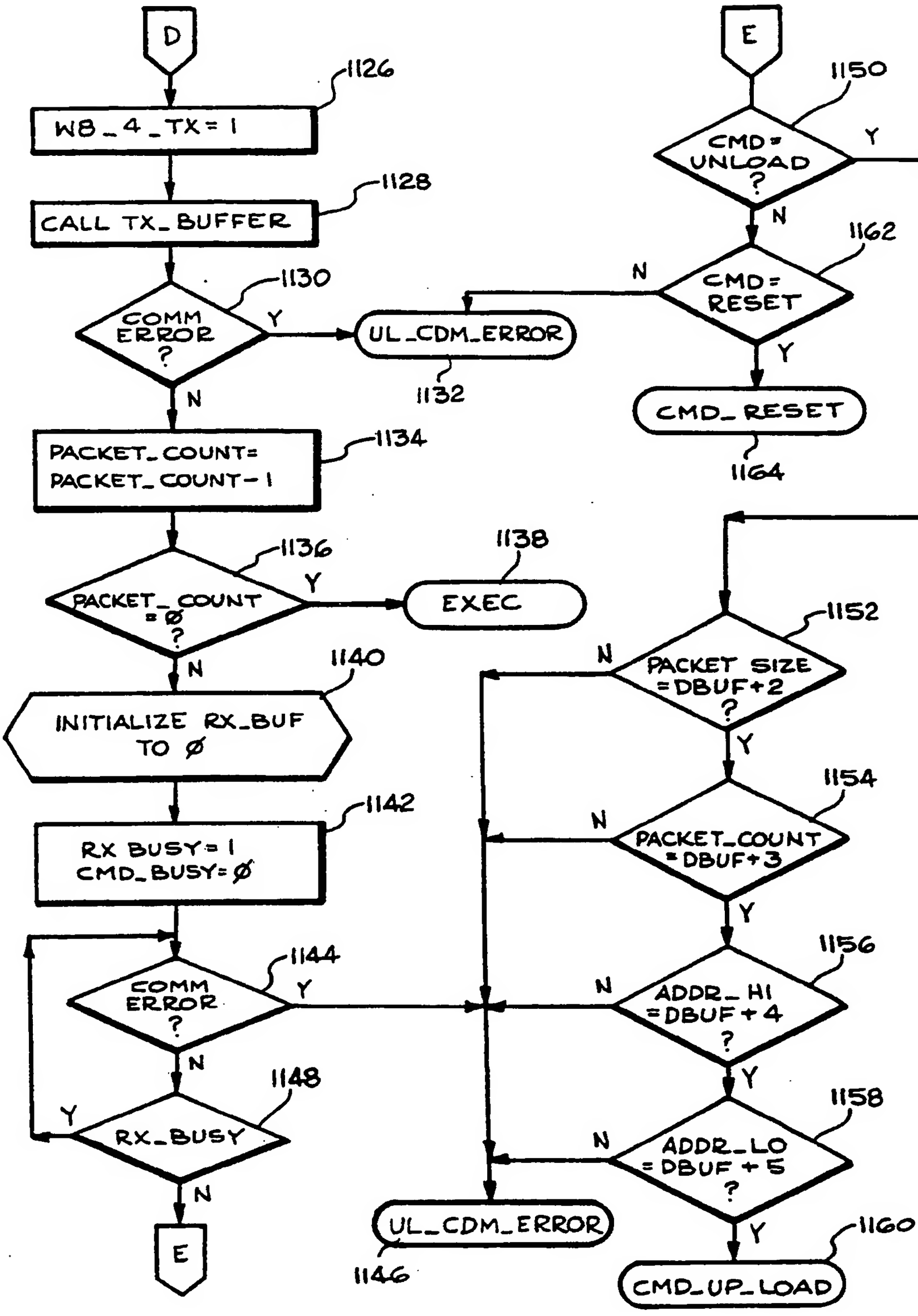
NSDOCID: <WO_____9325950A1_I_>

FIG.19



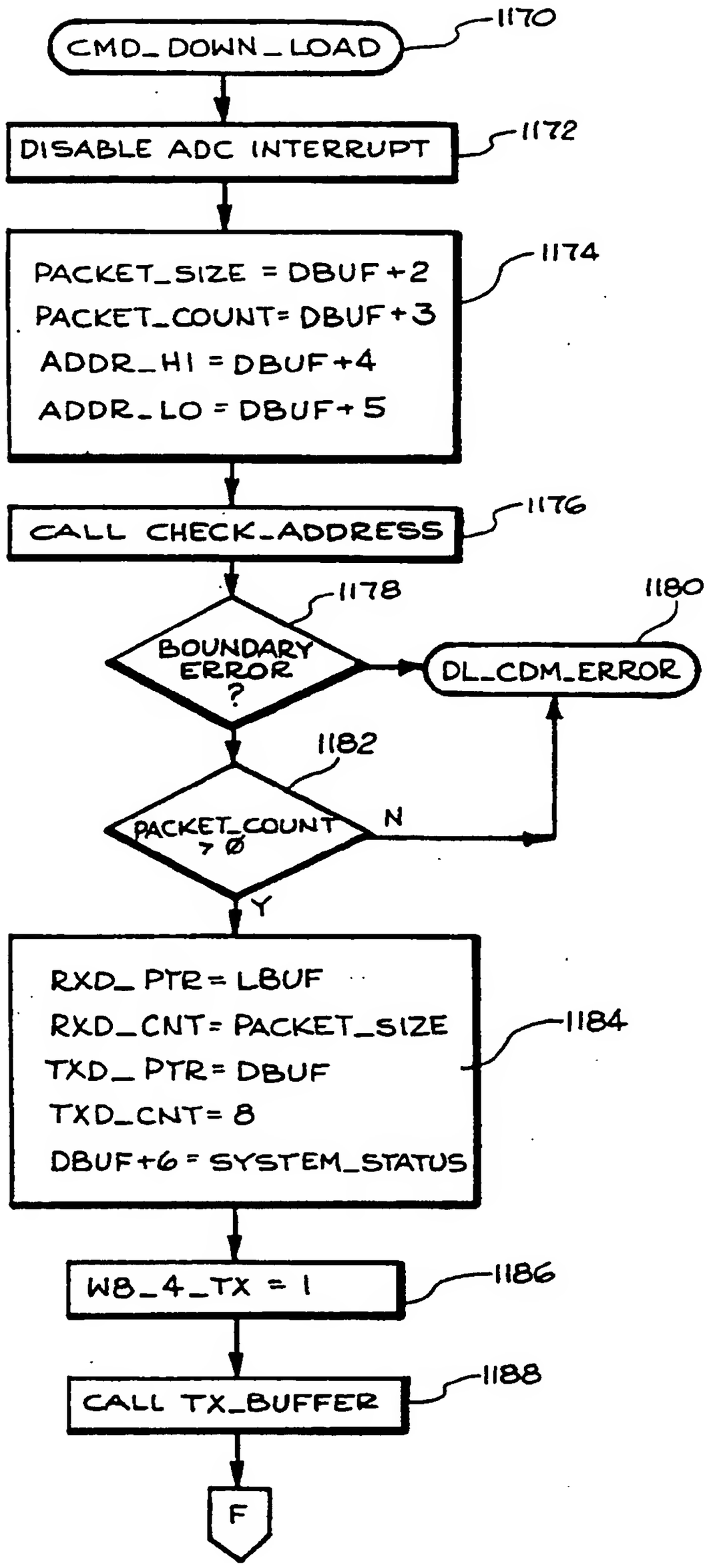
3NSDOCID: <WO_____9325950A1_I_>

FIG. 20



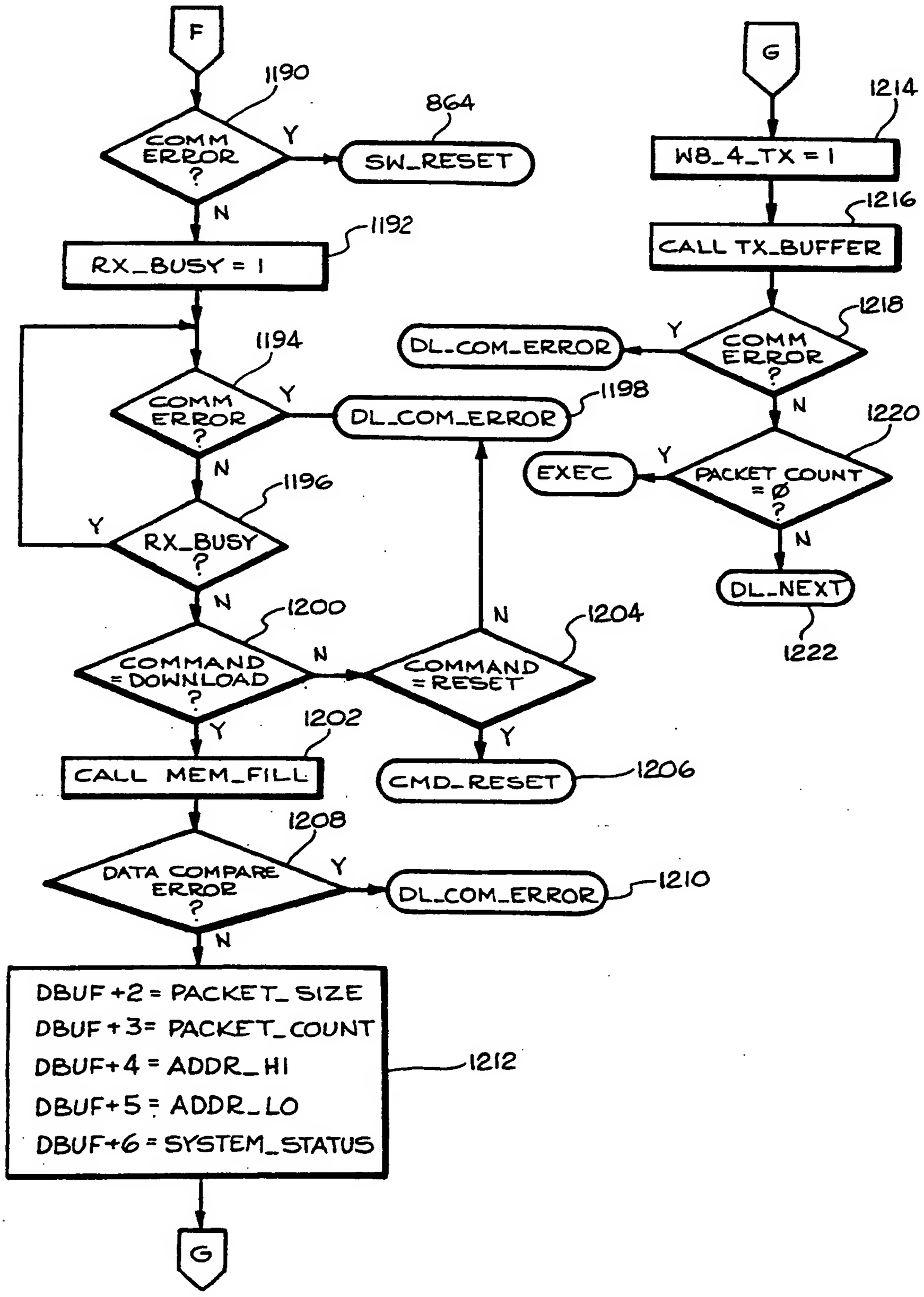
INSDOCID: <WO_____9325950A1_I_>

FIG. 21



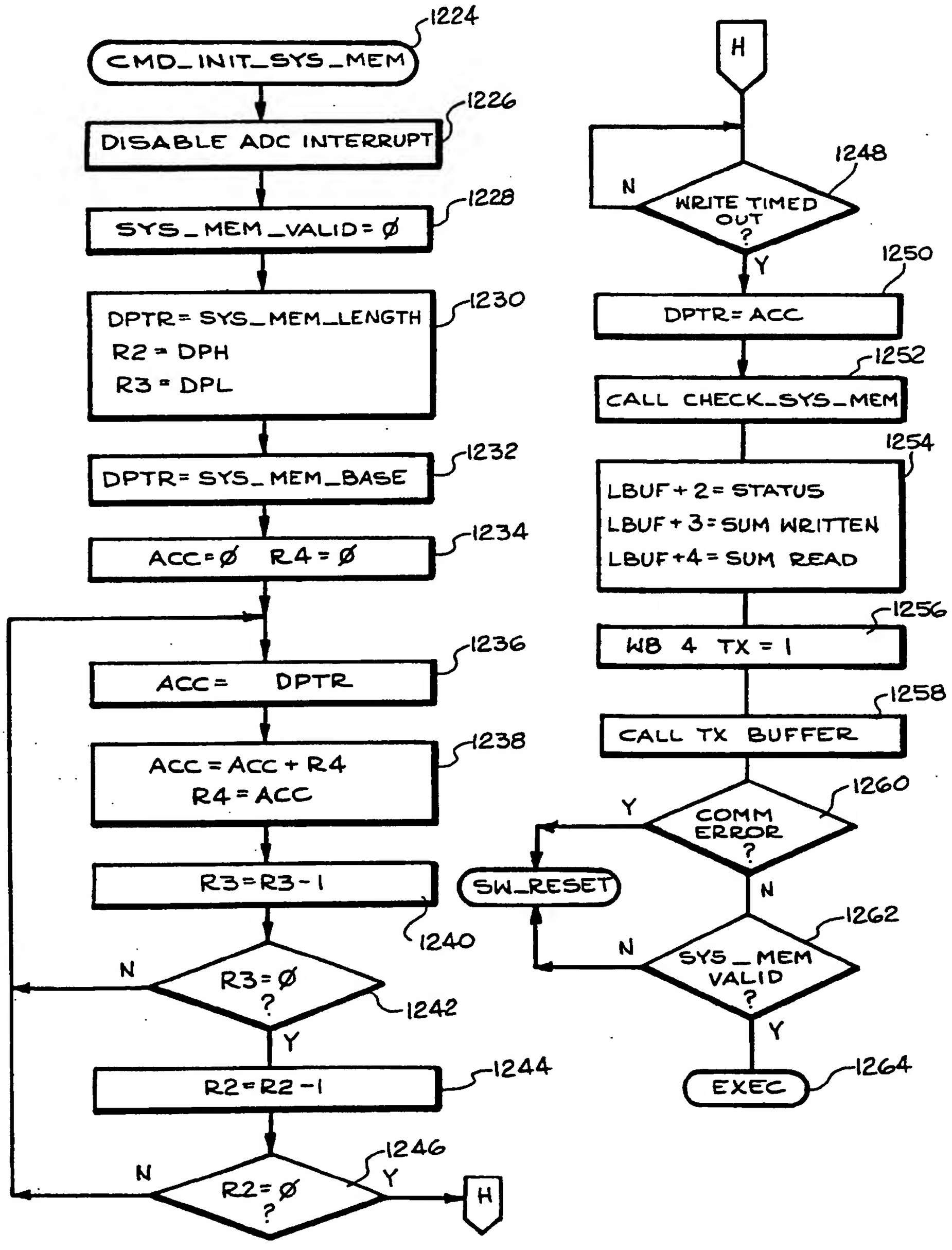
3NSDOCID: <WO_____9325950A1_I_>

FIG. 22



NSDOCID: <WO_____9325950A1_I_>

FIG. 23



3NSDOCID: <WO_____9325950A1_I_>

FIG. 24A

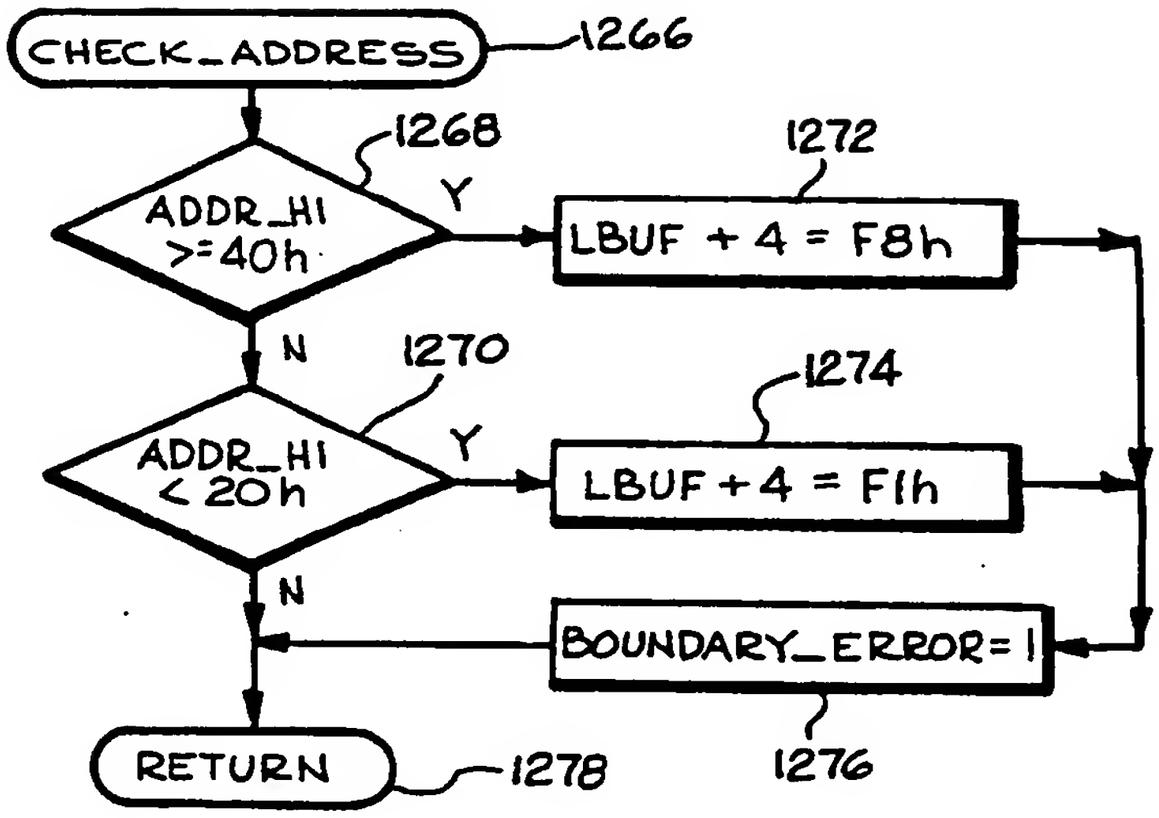
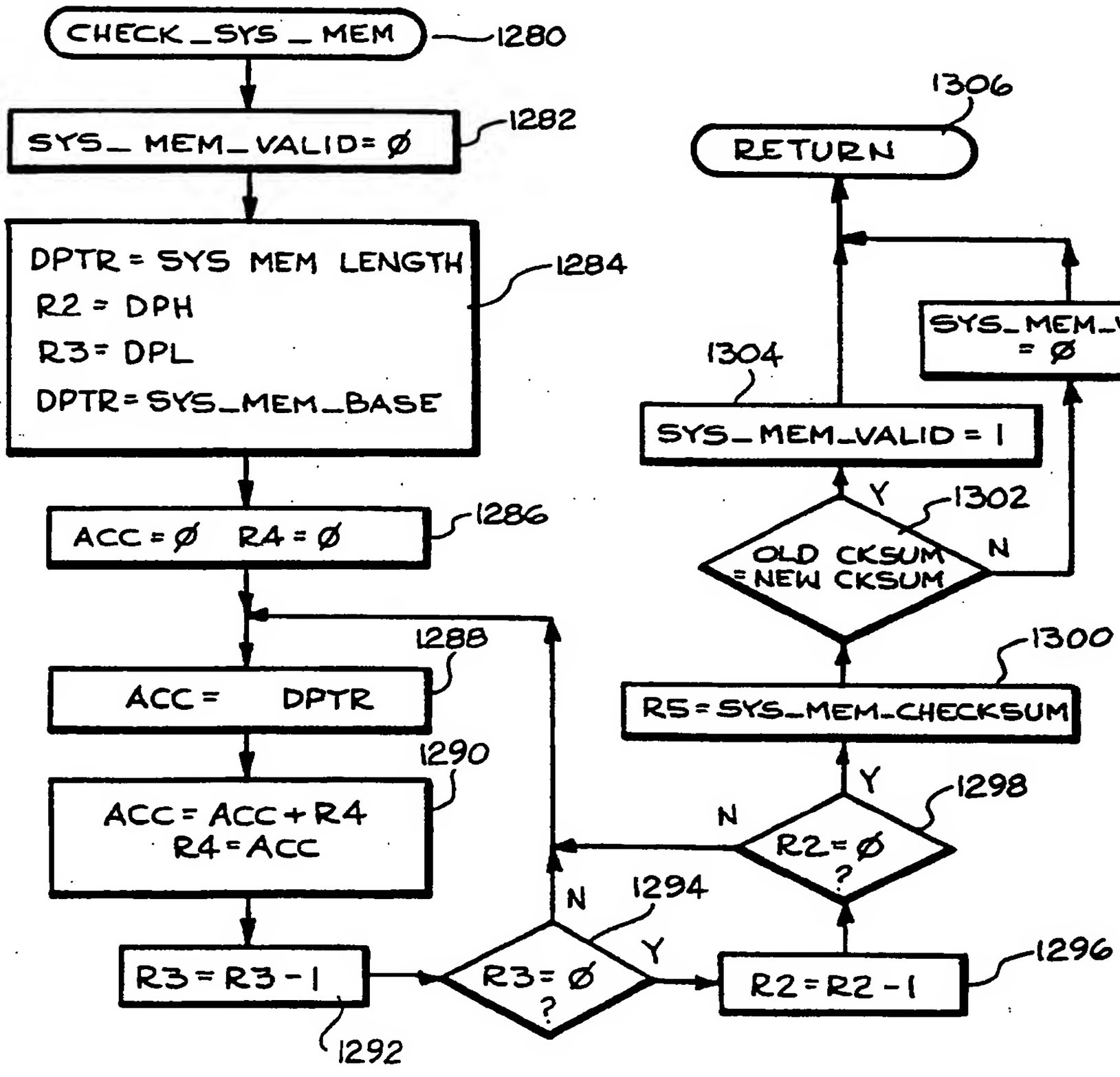
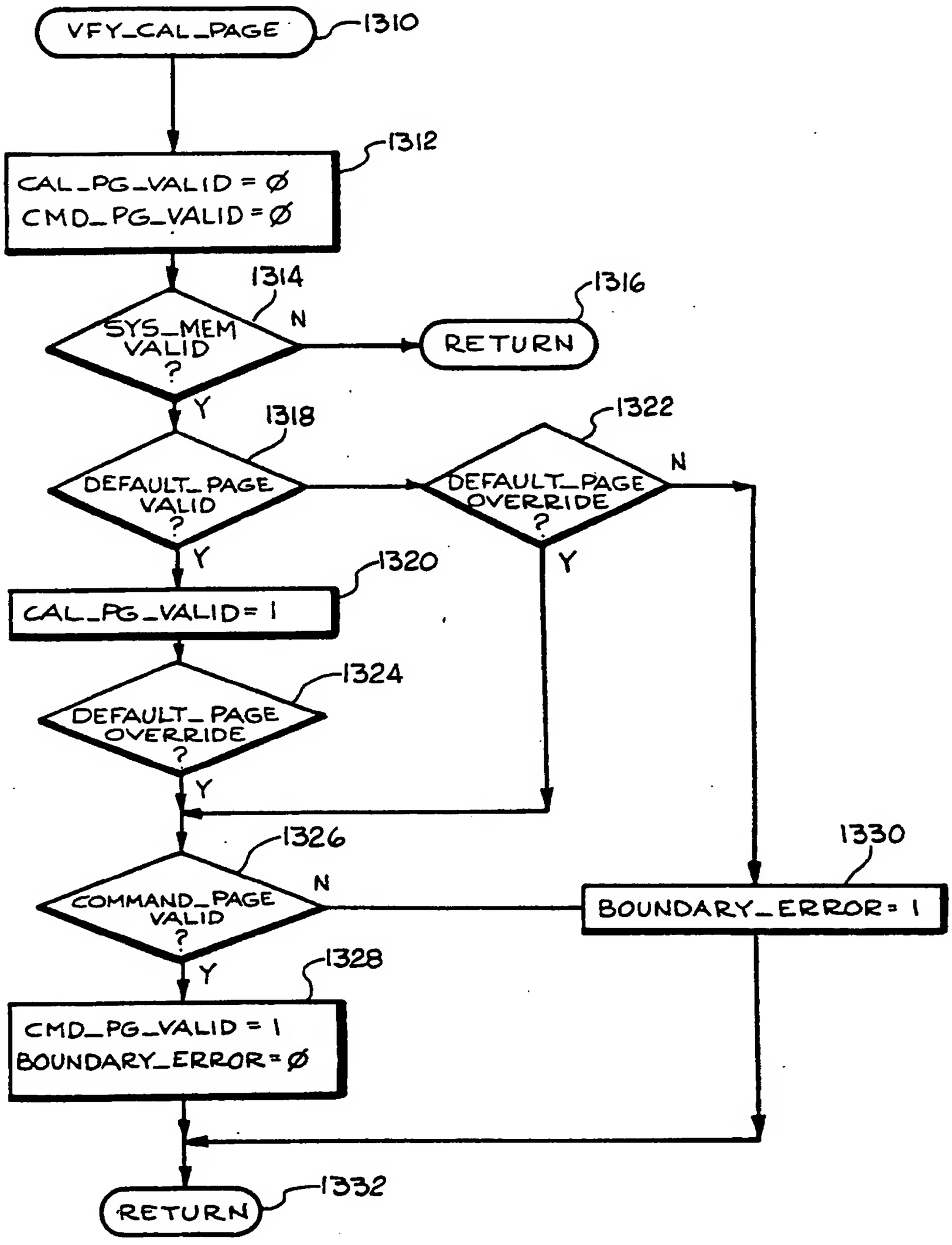


FIG. 24B



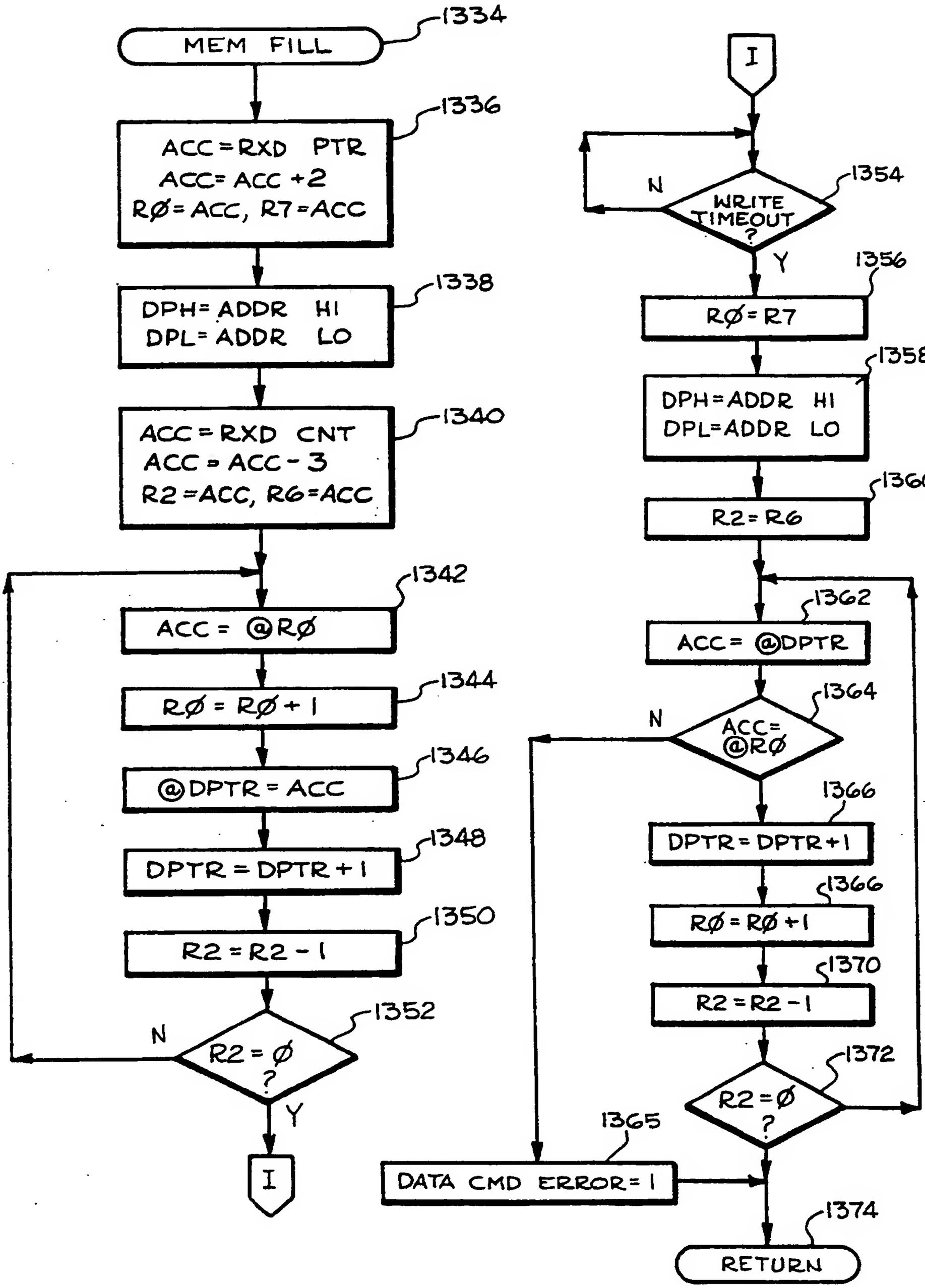
3NSDOCID: <WO_____9325950A1_I_>

FIG.25



3NSDOCID: <WO_____9325950A1_I_>

FIG. 26



NSDOCID: <WO_____9325950A1_I_>

FIG. 27A

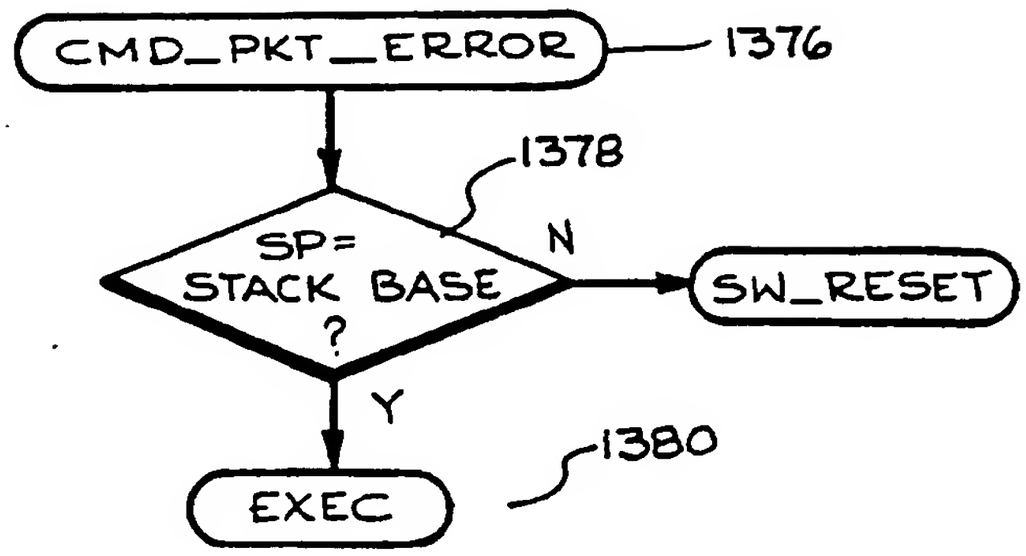
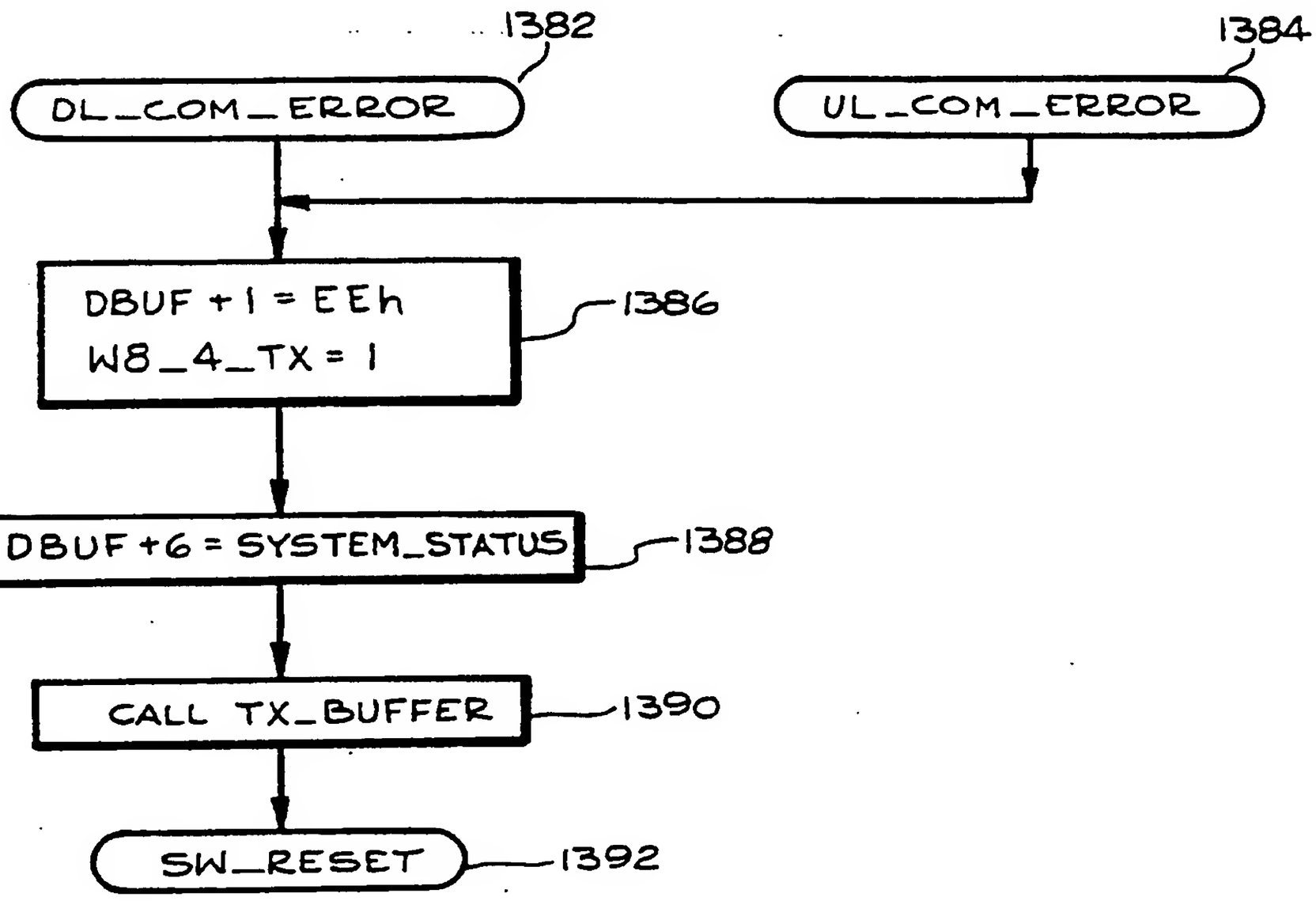
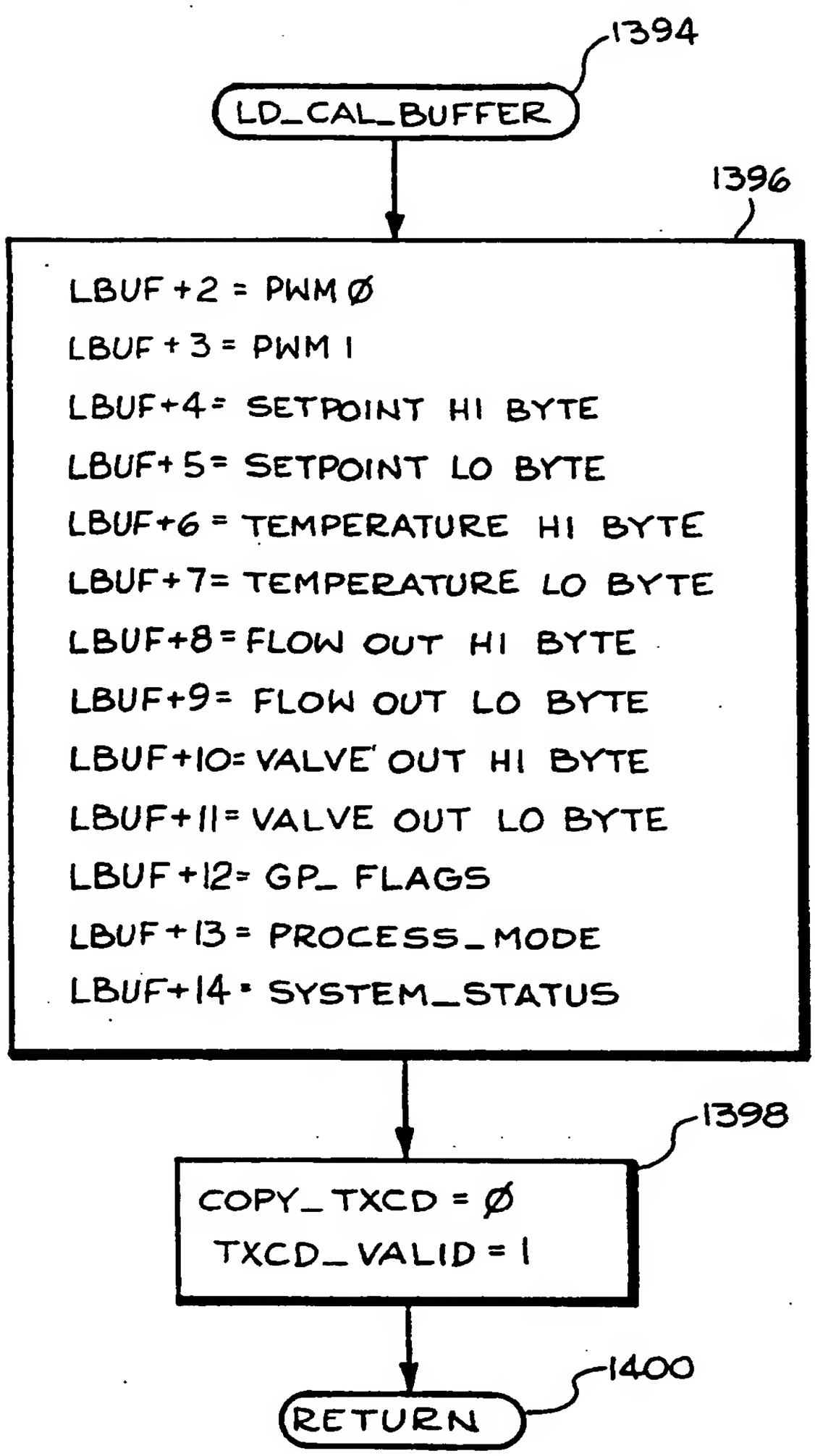


FIG. 27B



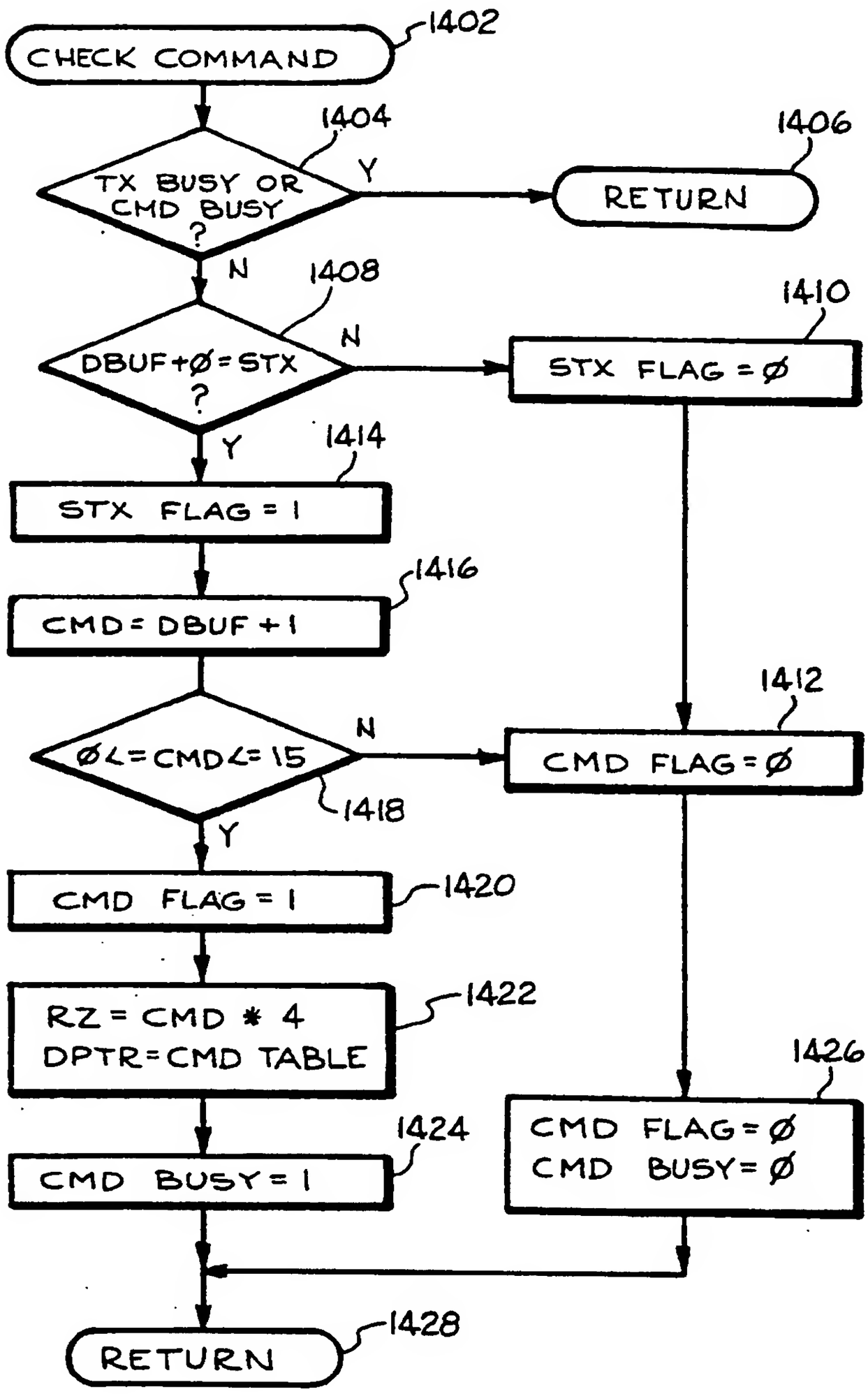
INSDOCID: <WO_____9325950A1_I_>

FIG. 28



INSDOCID: <WO_____9325950A1_I_>

FIG. 29



NSDOCID: <WO_____9325950A1_I_>

FIG. 30A

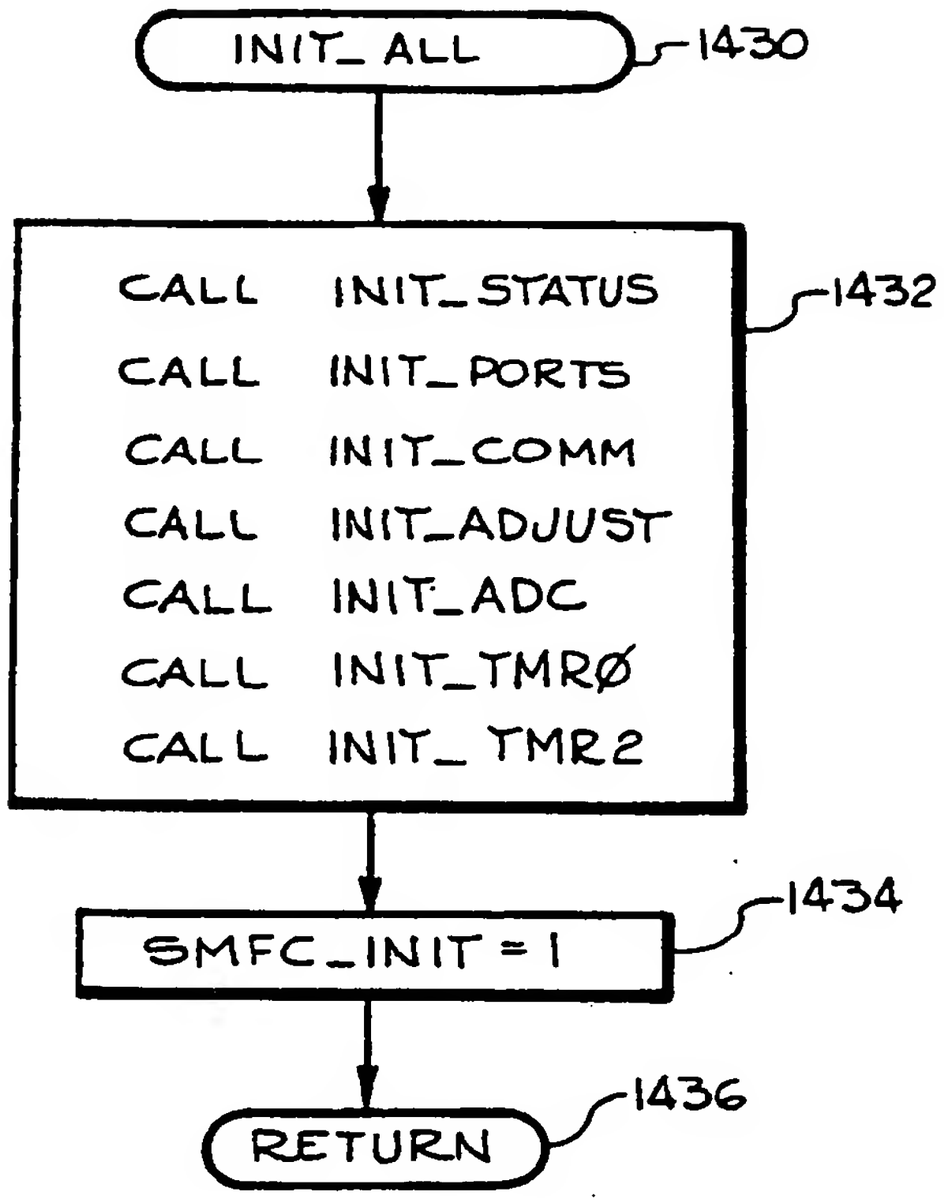


FIG. 30B

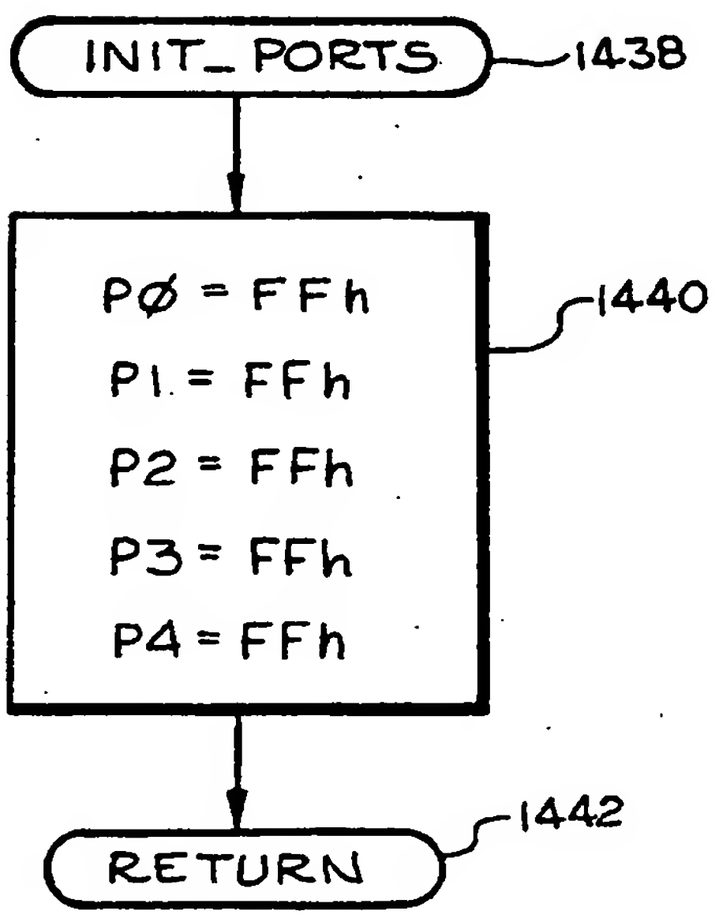


FIG. 30C

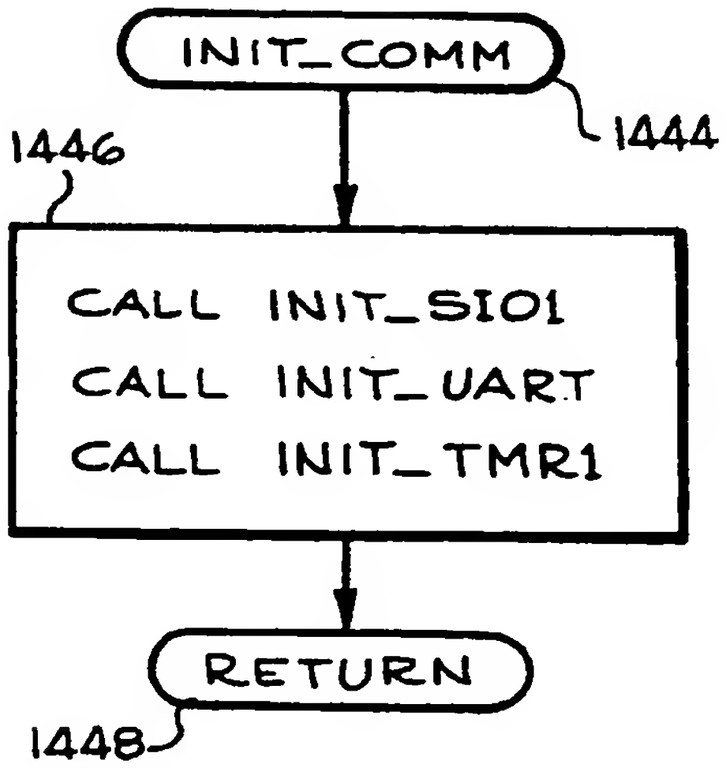
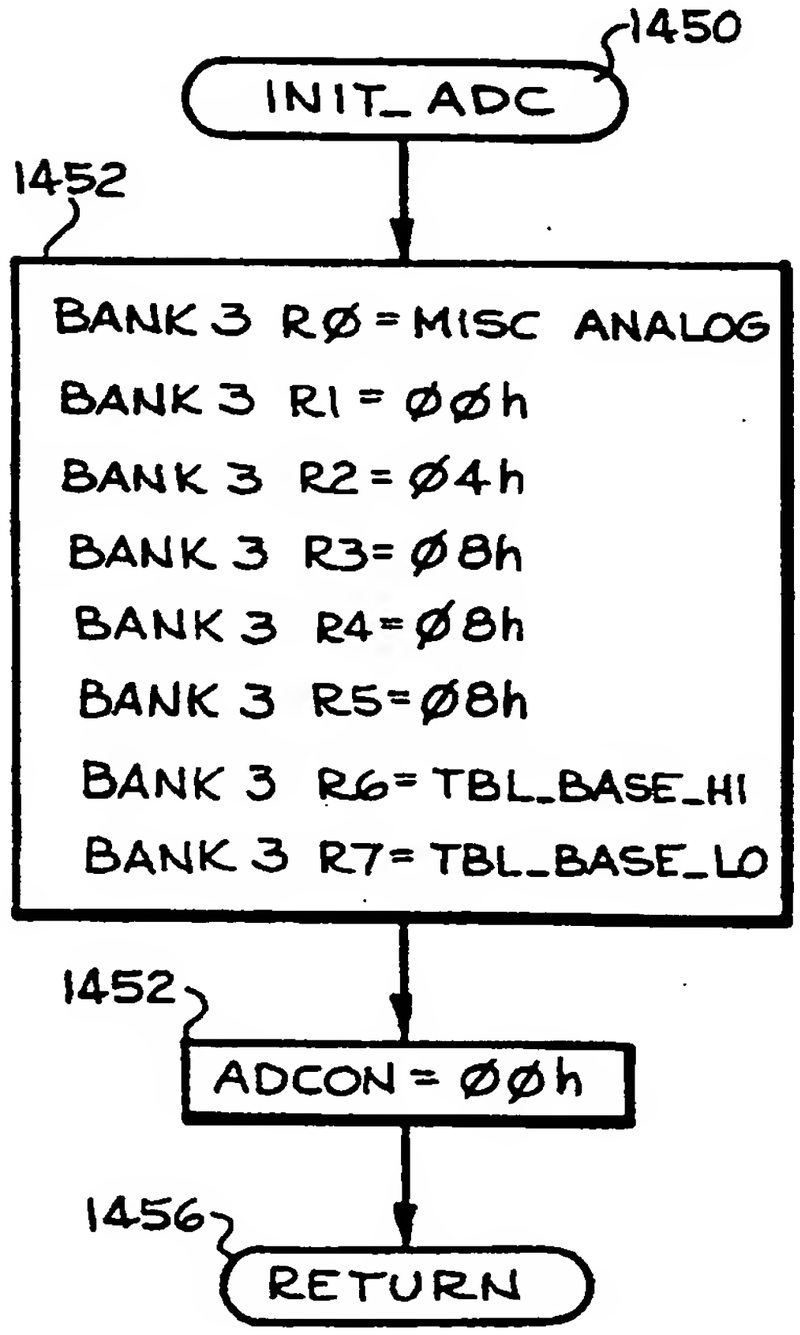
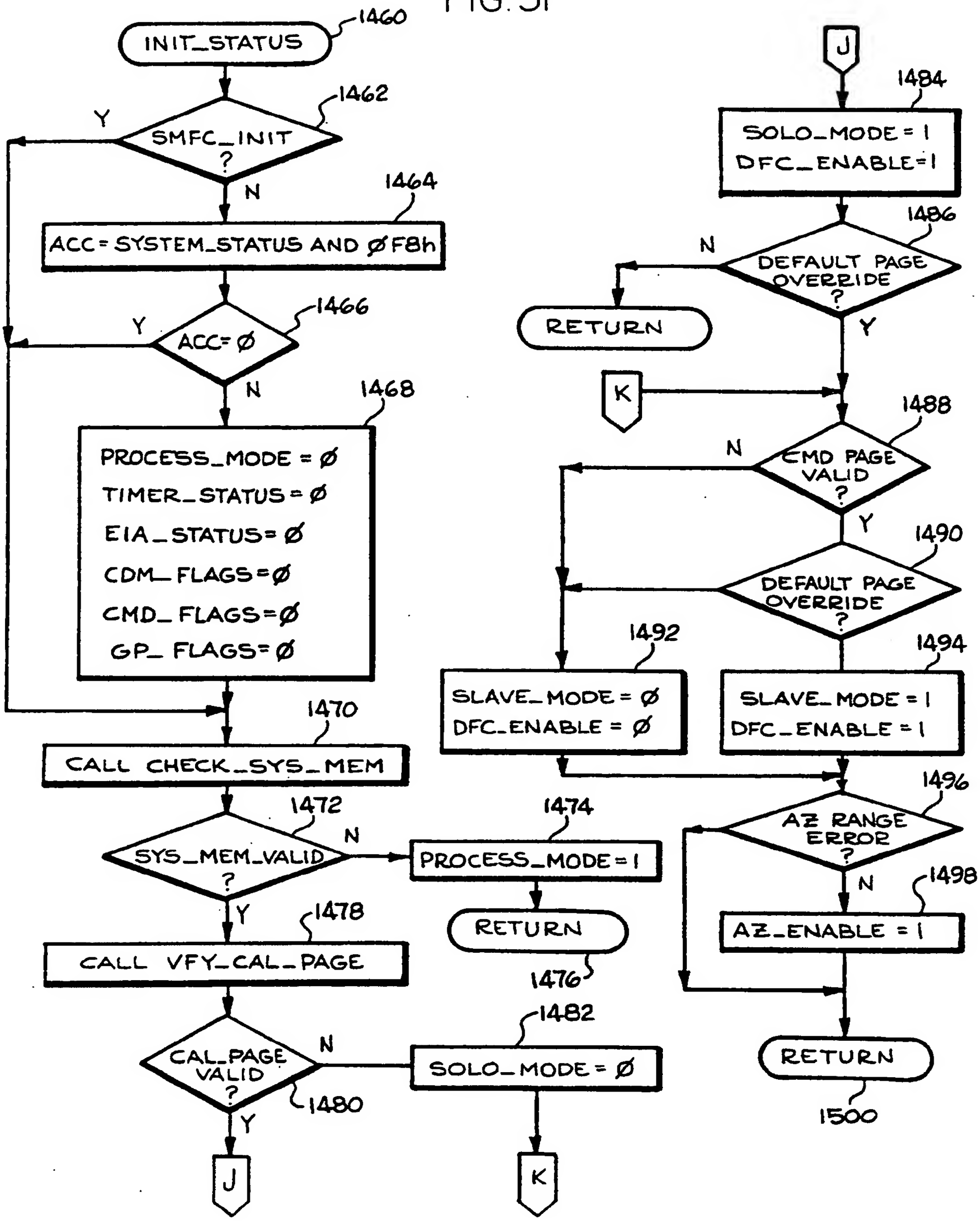


FIG. 30D



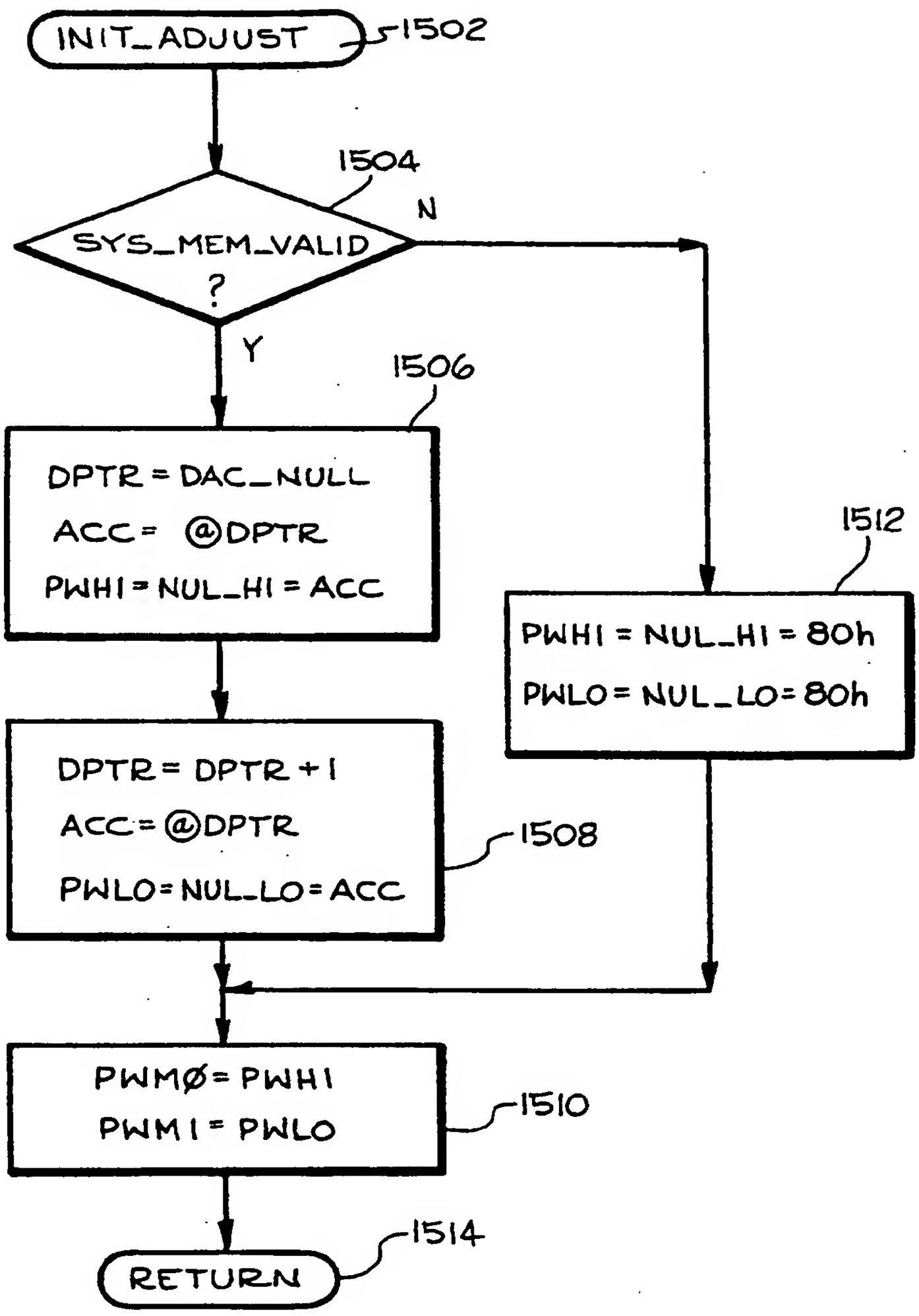
INSDOCID: <WO_____9325950A1_I_>

FIG. 31



3NSDOCID: <WO_____9325950A1_I_>

FIG.32



NSDOCID: <WO_____9325950A1_I_>

FIG. 33

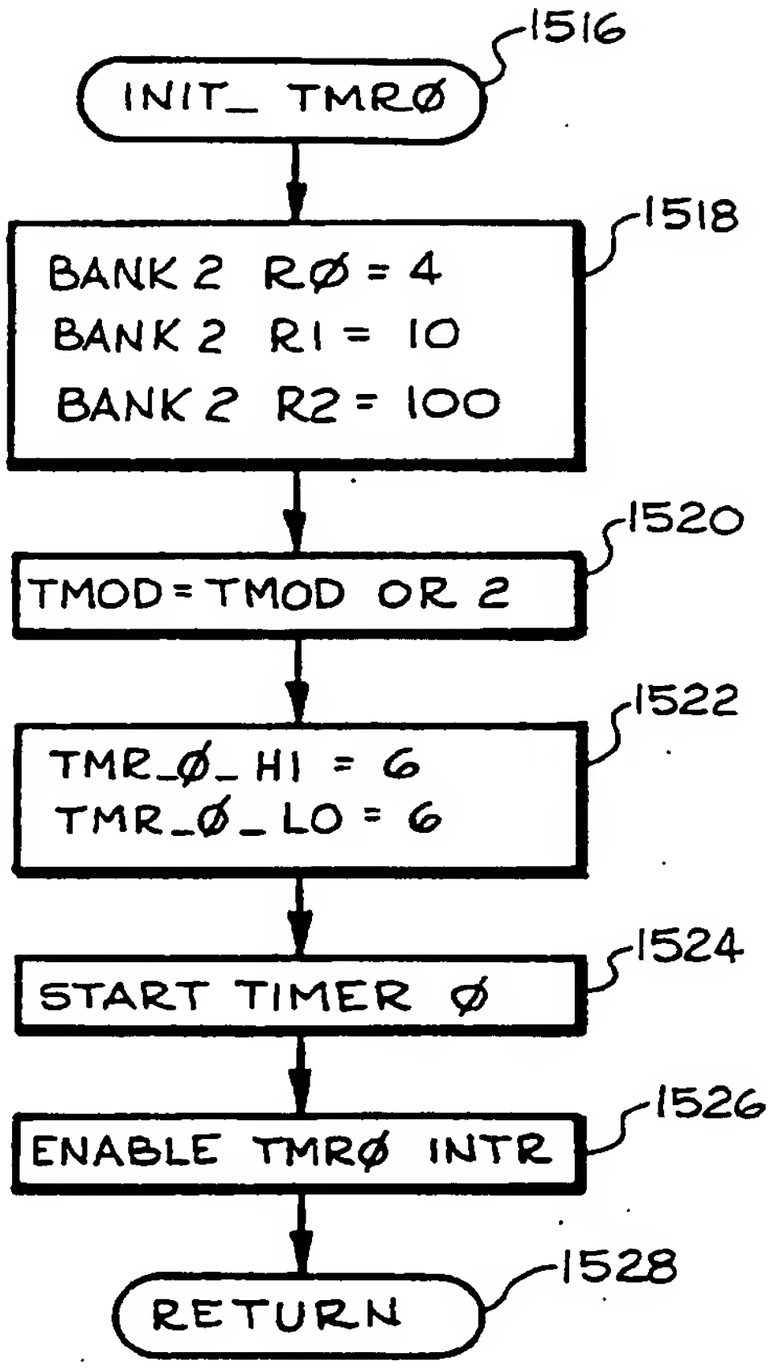


FIG. 34

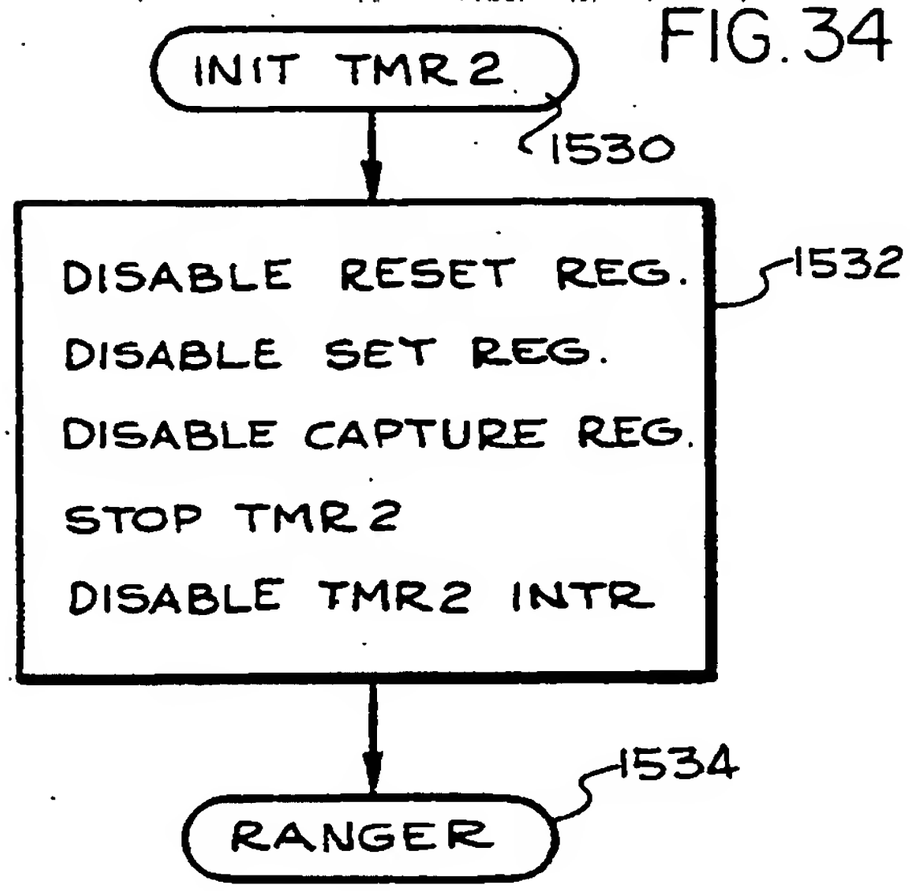


FIG. 35

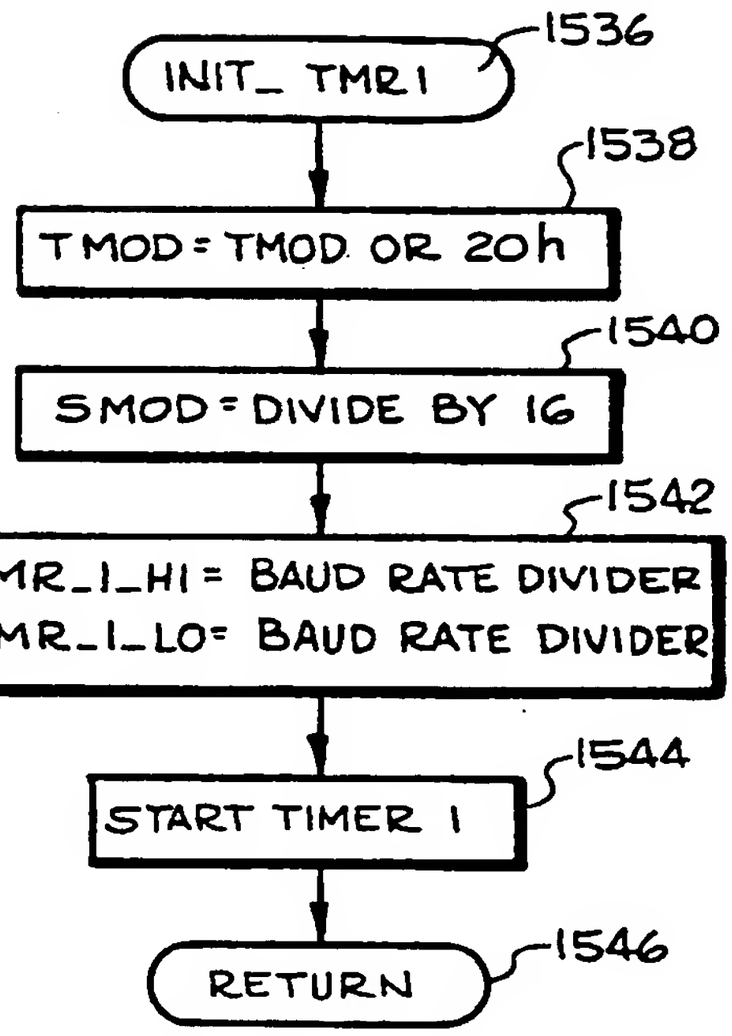
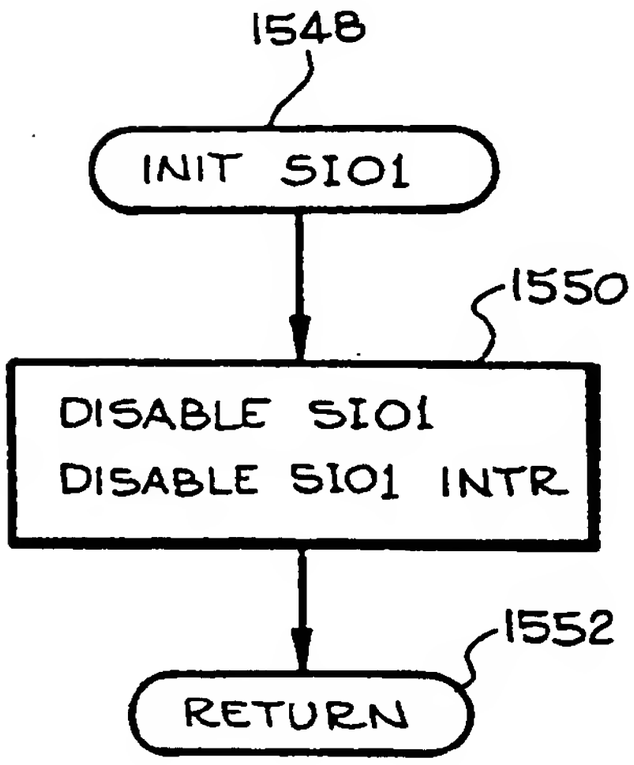


FIG. 36



NSDOCID: <WO_____9325950A1_I_>

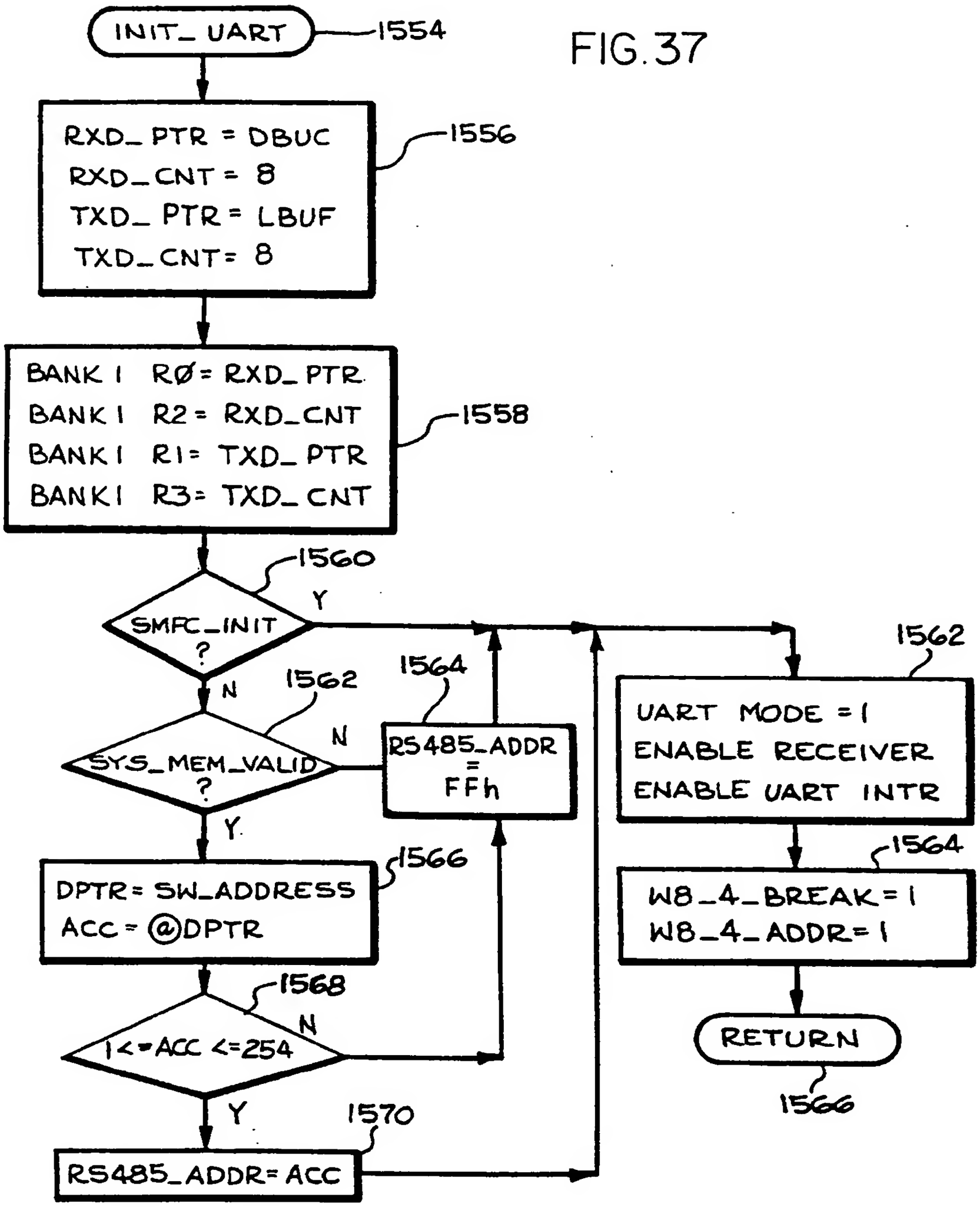


FIG. 37

3NSDOCID: <WO_____9325950A1_I_>

FIG.38

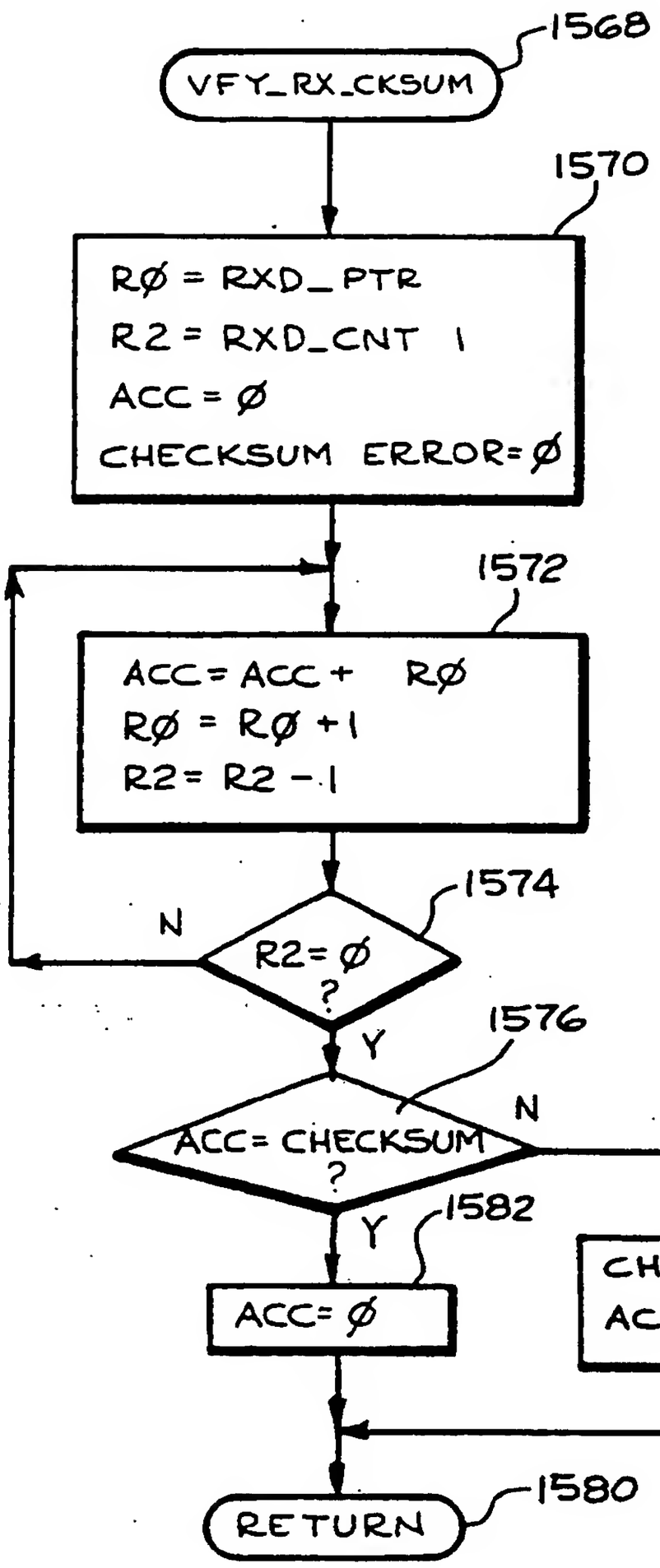
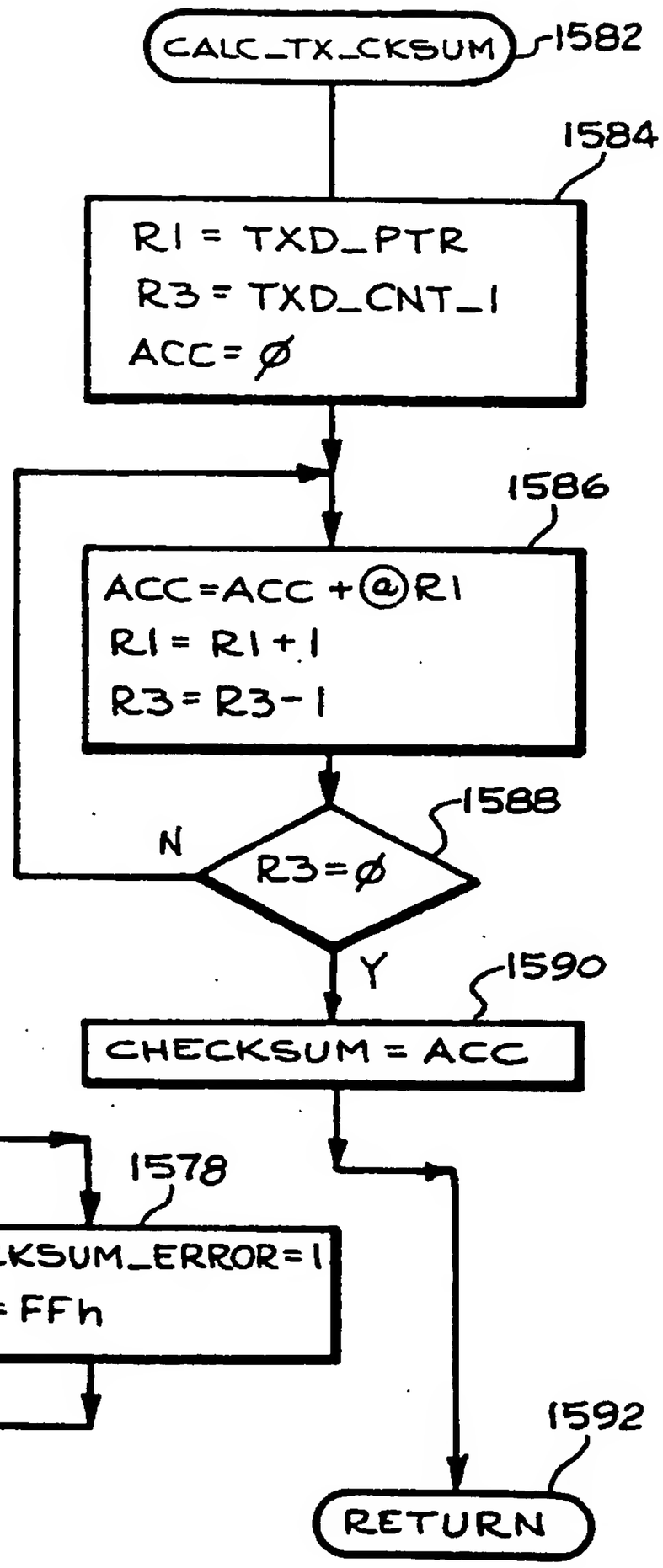
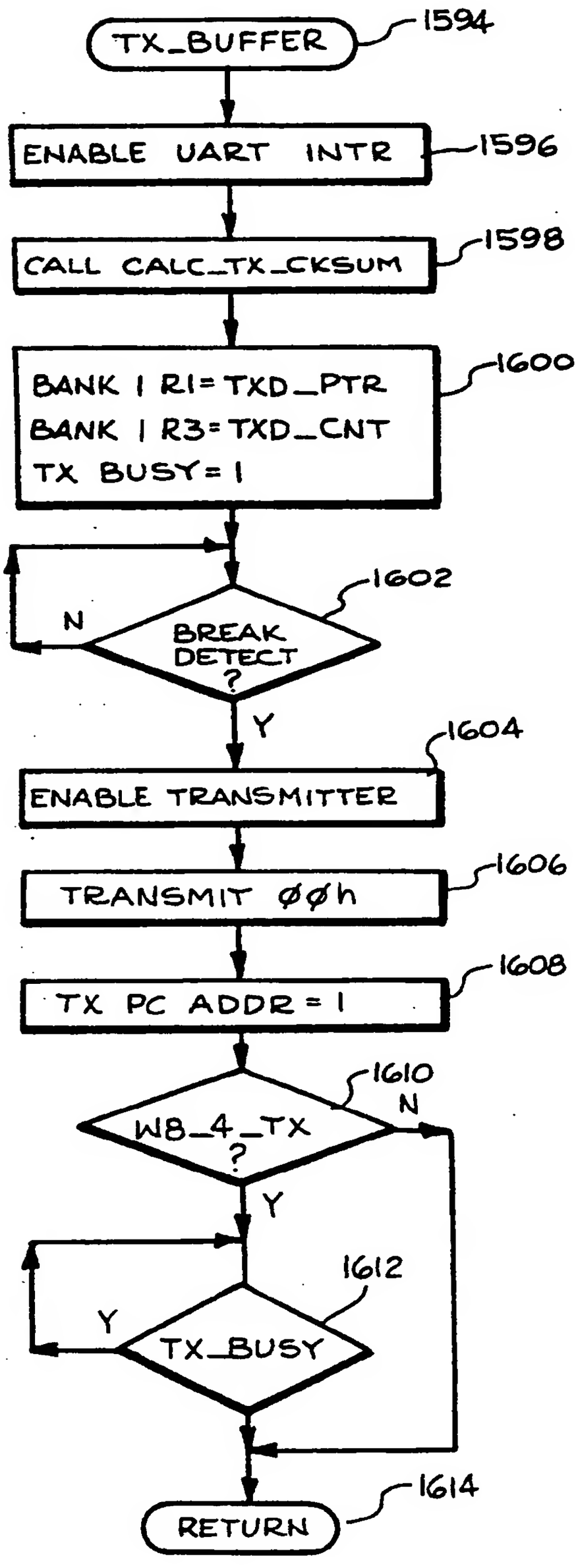


FIG.39



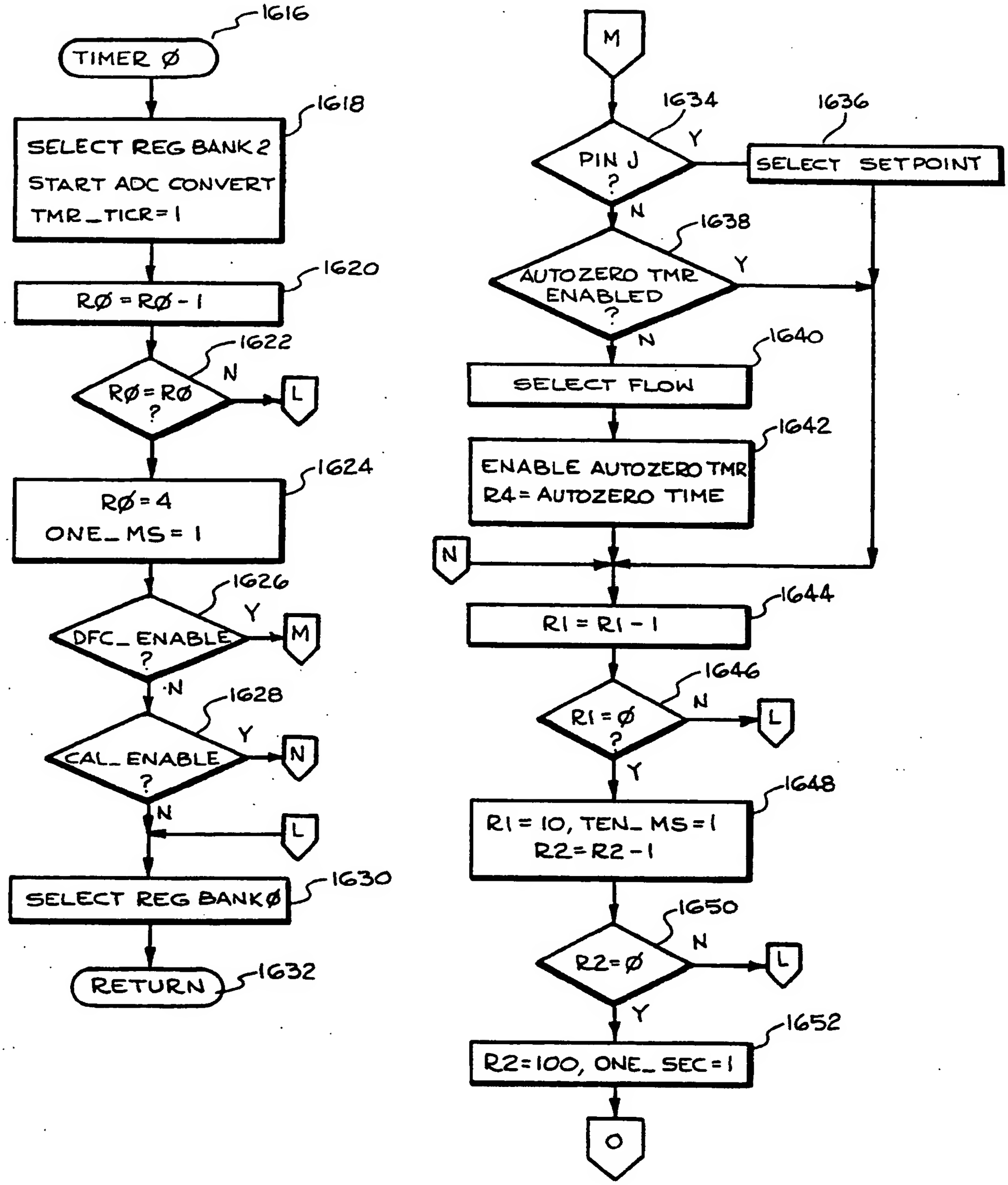
INSDOCID: <WO_____9325950A1_I_>

FIG.40



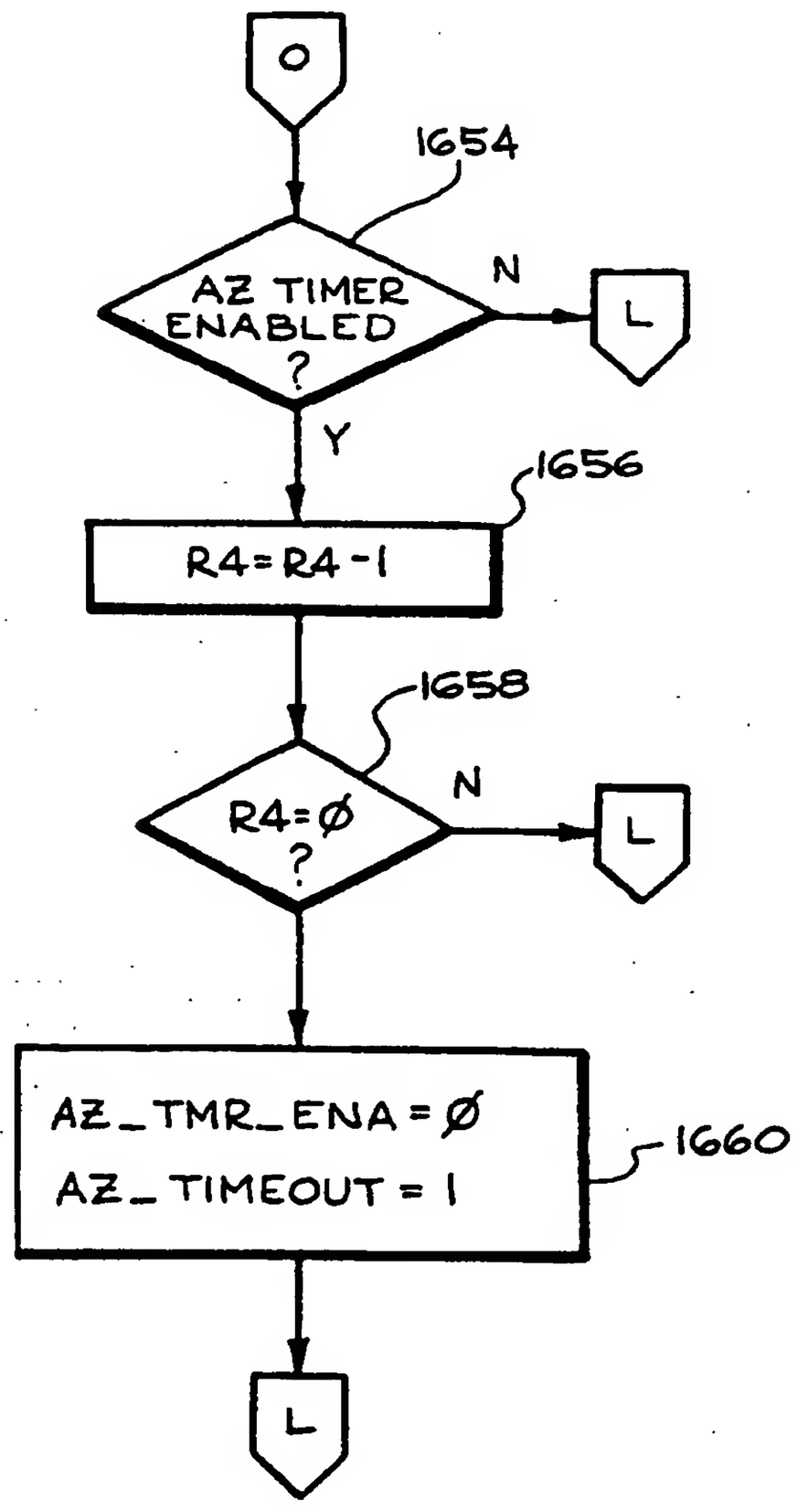
INSDOCID: <WO_____9325950A1_I_>

FIG.41A



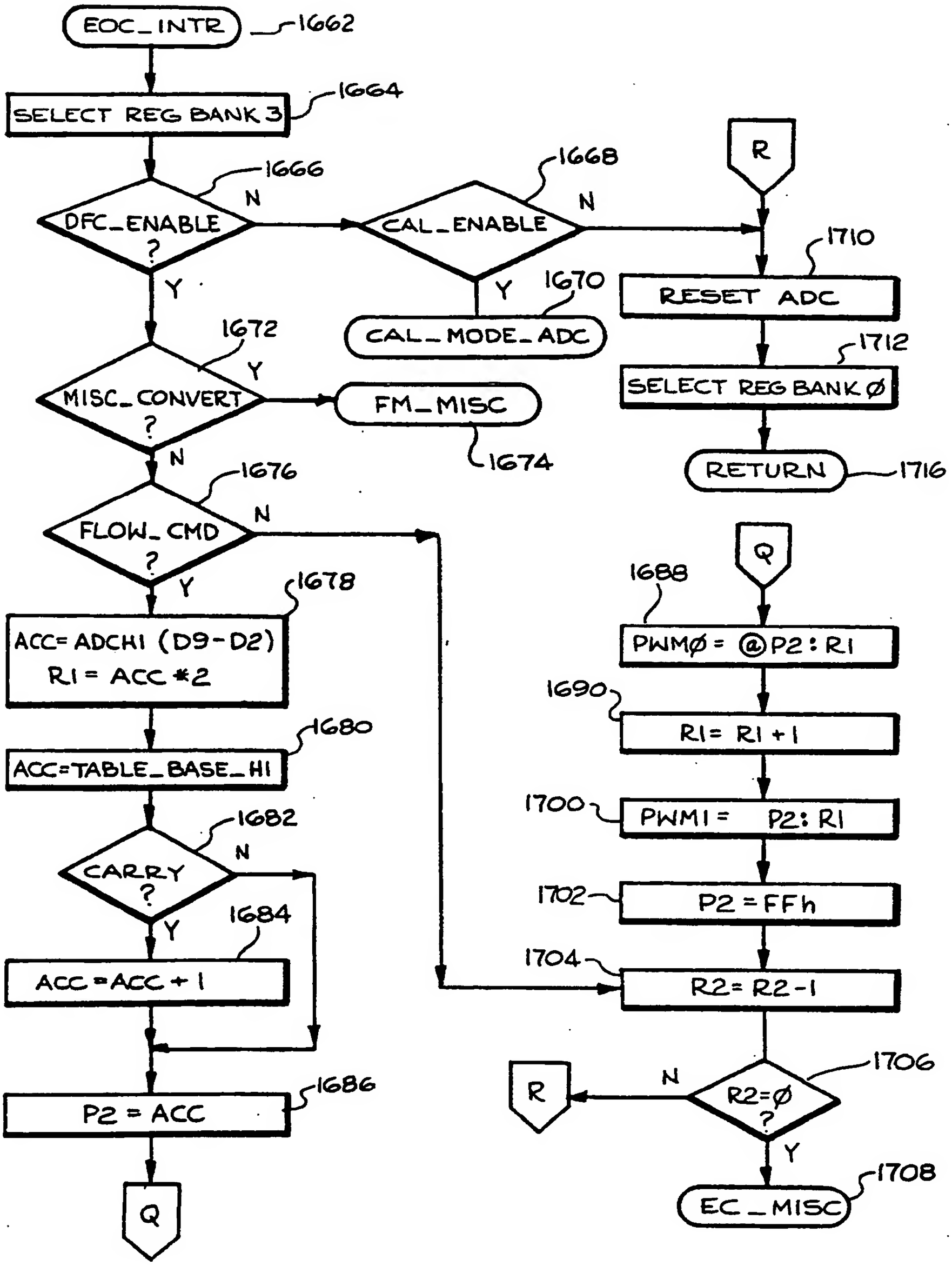
SDOCID: <WO_____9325950A1_I_>

FIG. 41B



3NSDOCID: <WO_____9325950A1_I_>

FIG.42



NSDOCID: <WO_____9325950A1_I_>

FIG.43

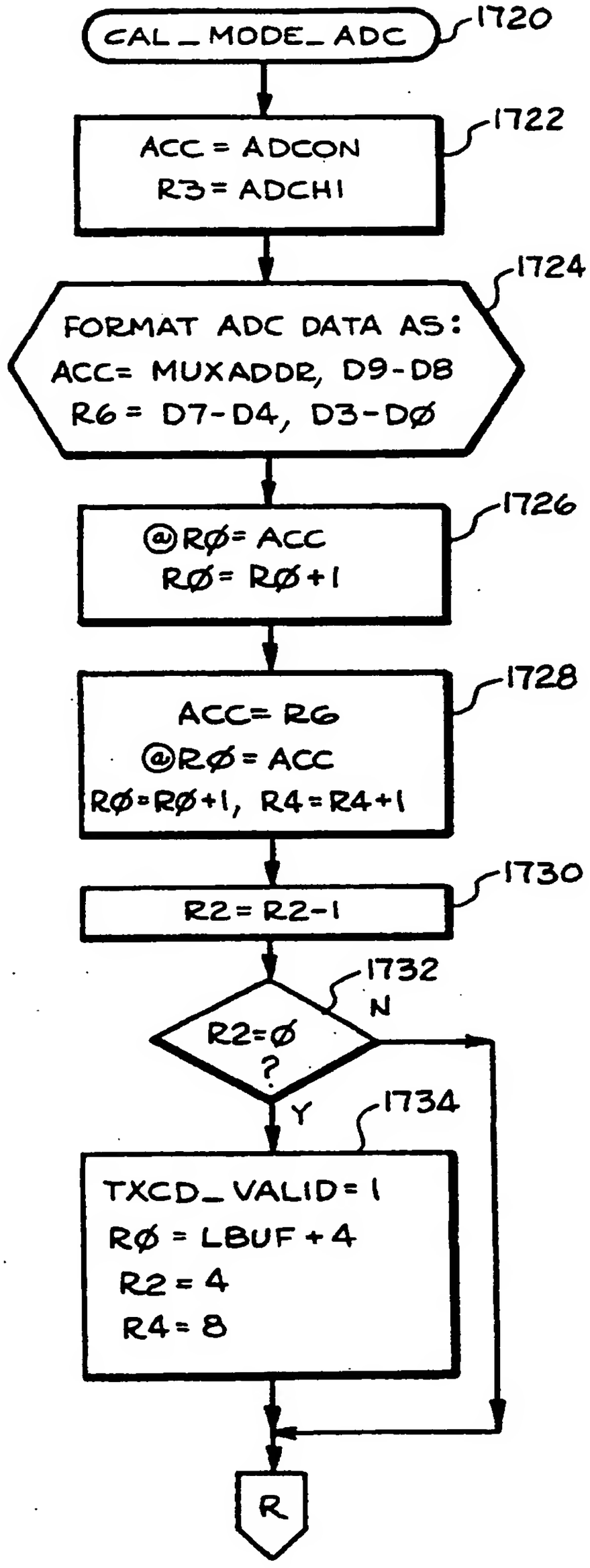
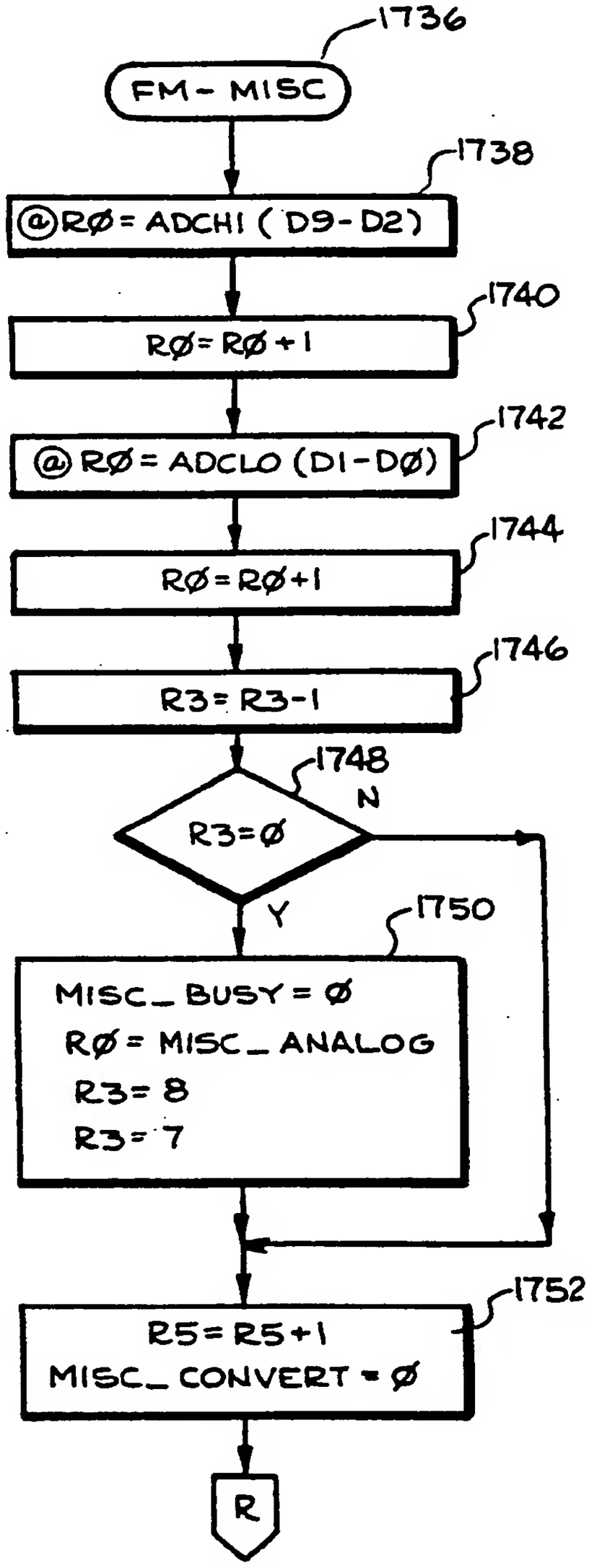


FIG.44



INSDOCID: <WO_____9325950A1_I_>

FIG.45

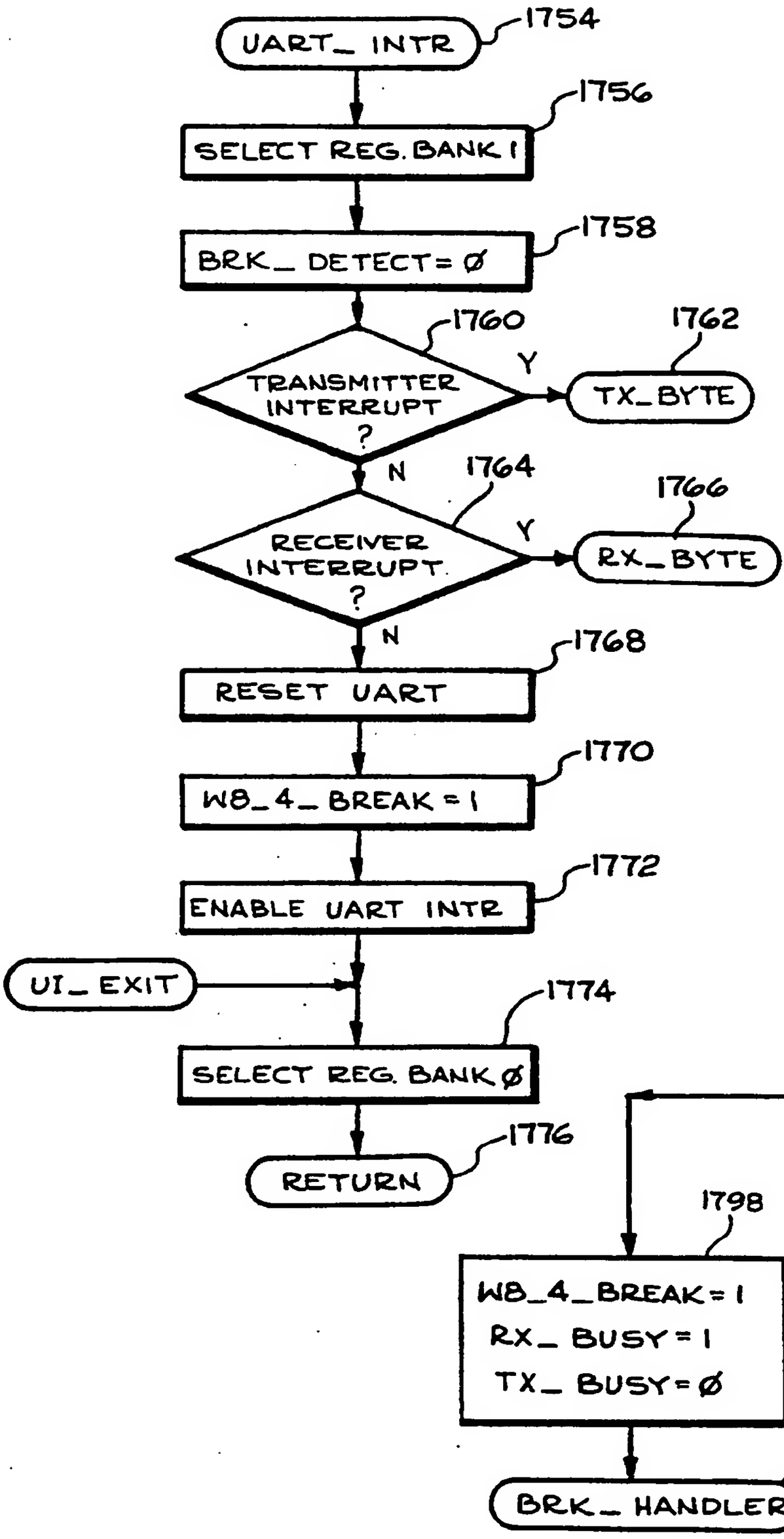
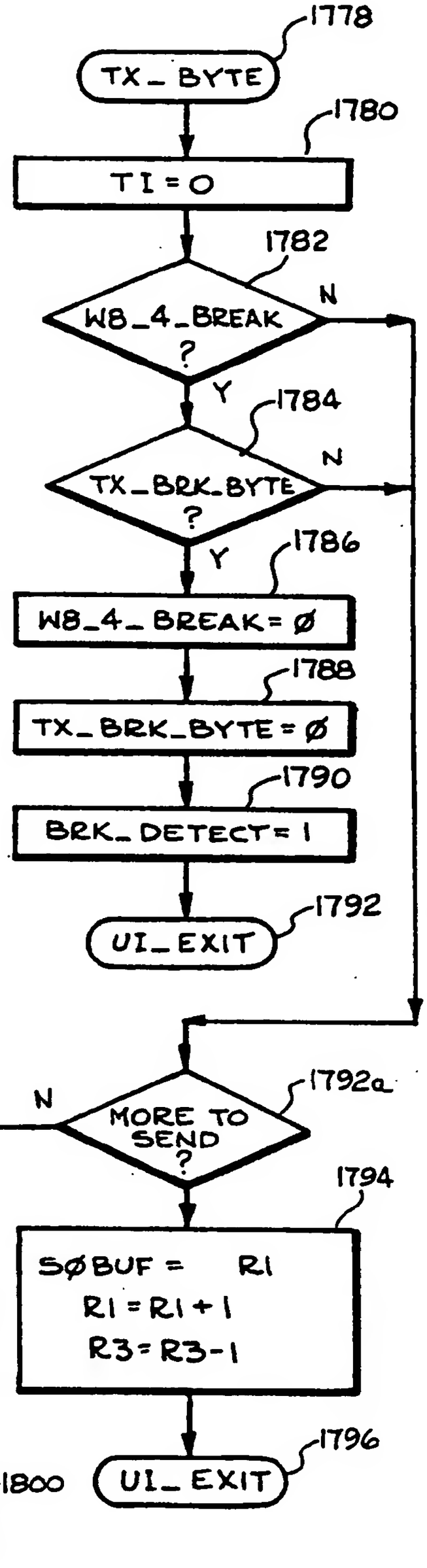
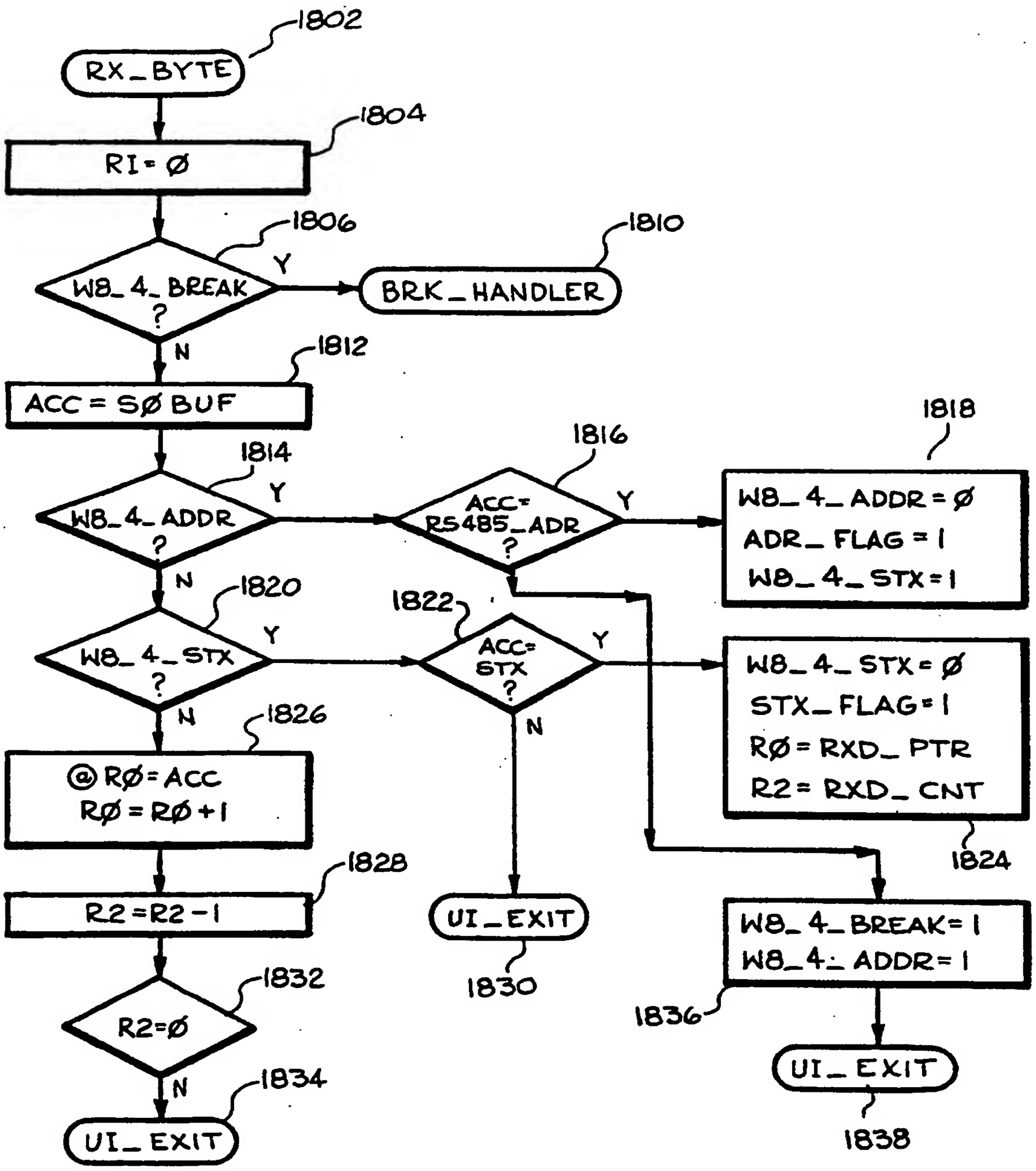


FIG.46



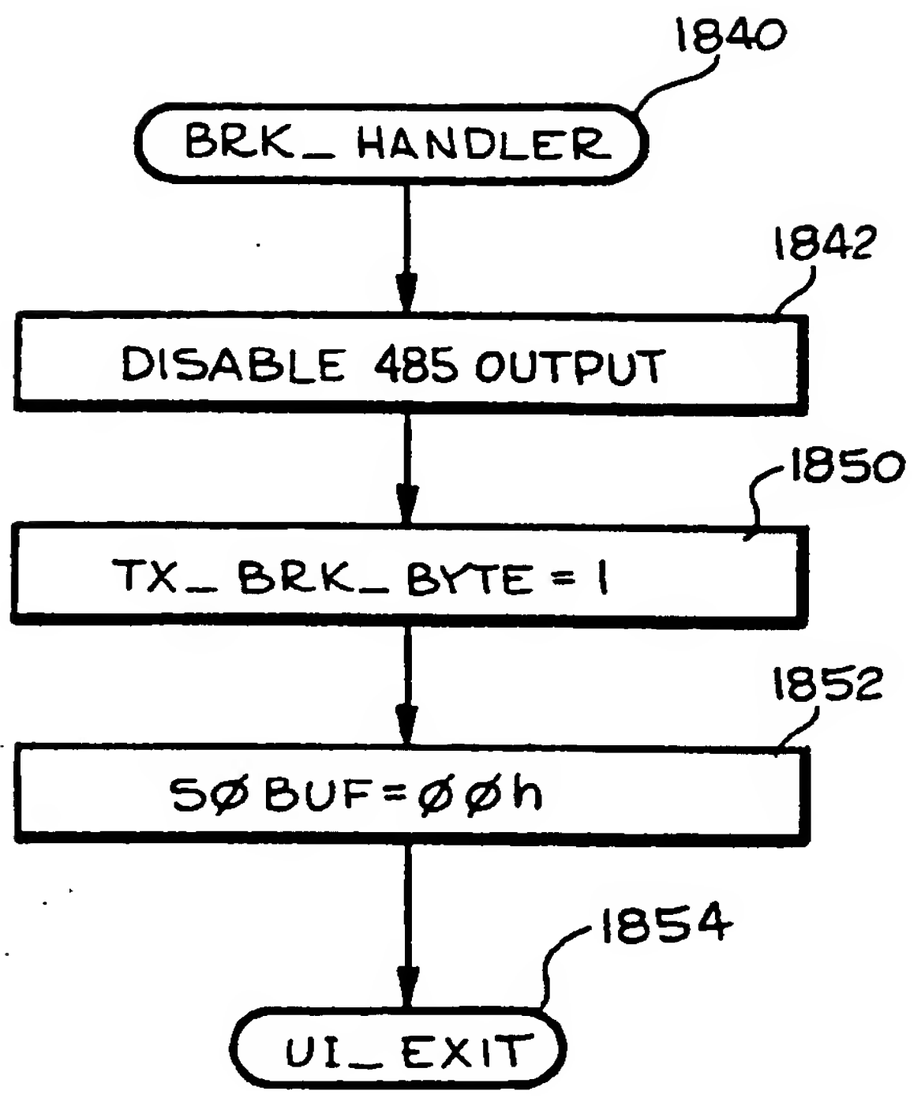
INSDOCID: <WO_____9325950A1_I_>

FIG.47



NSDOCID: <WO_____9325950A1_I_>

FIG. 48



NSDOCID: <WO_____9325950A1_I_>

FIG. 49

COMMAND VECTOR TABLE

COMMAND	VECTOR
1	CMD_RESET
2	CMD_DFC_ENTRY
3	CMD_TX_STATUS
4	CMD_SEL_TABLE
8	CMD_CALIBRATE
9	CMD_UP_LOAD
10	CMD_DOWN_LOAD
11	CMD_INIT_SYS_MEM
Ø, 5, 6, 7, 12, 13, 14, 15	SW_RESET

FIG. 50

EVENT VECTOR TABLE

EVENT	VECTOR
RESET	SMFC_START
EXTERNAL INTERRUPT Ø	XIOSVC
TIMER Ø OVERFLOW	TIMER_Ø
EXTERNAL INTERRUPT 1	XI1SVC
TIMER 1 OVERFLOW	TM1SVC
SIO Ø (UART) INTERRUPT	UART_INTR
SIO 1 (I^2C) INTERRUPT	I2CSVC
TIMER 2 CAPTURE (Ø-3) INTERRUPT	T2CPT
ADC END OF CONVERSION	EOC_INTR
TIMER 2 COMPARE (Ø-2) INTERRUPT	T2CMP
TIMER 2 OVERFLOW	TM2SVC

INSDOCID: <WO_____9325950A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US93/05542

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G05D 16/20
US CL :137/486

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 137/486, 487.5, 624.11

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 4,487,213 (Gates et al) 11 December 1984, see Figure 2.	1-9
A	US, A, 4,658,855 (Doyle) 21 April 1987	
A,P	US, A, 5,141,021 (Shimomura et al) 25 August 1992	
A	US, A, 4,877,051 (Day) 31 October 1989	
A	US, A, 4,921,005 (Ohmi et al) 01 May 1990	
A	US, A, 5,100,100 (Benson et al) 31 March 1992	
A	US, A, 5,080,131 (Ono et al) 14 January 1992	

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 AUGUST 1993	19 AUG 1993
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE	Authorized officer GERALD A. MICHALSKY Telephone No. (703) 308-1049

Form PCT/ISA/210 (second sheet)(July 1992)★

3NSDOCID: <WO_____9325950A1_I_>